INTELLIGENT BEHAVIORAL ACTION AIDING FOR IMPROVED AUTONOMOUS IMAGE NAVIGATION

THESIS

Kwee Guan Eng, ME4 (Captain), Republic of Singapore Air Force

AFIT/GE/ENG/12-46

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/12-46

INTELLIGENT BEHAVIORAL ACTION AIDING FOR IMPROVED AUTONOMOUS IMAGE NAVIGATION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Kwee Guan Eng

ME4 (Captain), Republic of Singapore Air Force

September 2012

AFIT/GE/ENG/12-46

INTELLIGENT BEHAVIORAL ACTION AIDING FOR IMPROVED AUTONOMOUS IMAGE NAVIGATION

Kwee Guan Eng
ME4 (Captain), Republic of Singapore Air Force

Approved:

//SIGNED//	4 September 2012
Dr Gilbert L. Peterson (Chairman)	Date
//SIGNED//	5 September 2012
Dr John F. Raquet (Member)	Date
//SIGNED//	5 September 2012
Lt Col Kenneth A. Fisher (Member)	Date

Abstract

In egomotion image navigation, errors are common especially when traversing areas with few landmarks. Since image navigation is often used as a passive navigation technique in Global Positioning System (GPS) denied environments, egomotion accuracy is important for precise navigation in these challenging environments. One of the causes of egomotion errors is inaccurate landmark distance measurement (e.g. sensor noise). This research develops a landmark location egomotion error model that quantifies the effects of landmark locations on egomotion value uncertainty and errors. The error model accounts for increases in landmark uncertainty due to landmark distance and image centrality. A robot then uses the error model to actively orient to position landmarks in image positions that give the least egomotion calculation uncertainty. Three action aiding solutions are proposed: (1) qualitative non evaluative aiding action, (2) quantitative evaluative aiding action with physical scans and (3) quantitative evaluative aiding action with landmark tracking. Simulation results show that action aiding techniques reduce the position uncertainty compared to no action aiding. Physical testing results substantiate simulation results. Compared to no action aiding, non evaluative action aiding reduced egomotion position errors by an average 31.5%, while evaluative action aiding reduced egomotion position errors by an average 72.5%. Physical testing also showed that evaluative action aiding enables egomotion to work reliably in areas with few features, achieving 76% egomotion position error reduction compared to no aiding.

To my wife who has supported and carried me through these times when I have spent countless hours on the research, away from her in my room or in school. To my son who has to spend many hours each day seeing his dad but not being able to play with or be entertained by dad as much as he would have liked. To my mum for coming with us here to Dayton Ohio to provide a helping hand in the house, freeing up much time for me to concentrate in my studies and research. To my dad and brother for staying back in Singapore without my mum and supporting my studies and research.

Acknowledgments

First and foremost, I like to thank my thesis advisor Dr Peterson for his guidance throughout the project without which, the successes in this research would not have been possible. I would also like to thank him for reviewing through this thesis numerous times. If not for the comments, suggestions and corrections, this thesis would have been much harder to follow and understand. I would also like to express my appreciation to the committee members Dr Raquet and Lt Col Fisher for showing support and much needed encouragement. I am also deeply grateful to the staff of the Advanced Navigation Technology (ANT) Center especially Mr Kresge who worked tirelessly to help integrate the action aiding engine into the robot for testing. The support of AFRL has been crucial as the sponsor for this research. I also like to thank my lab mates James who reviewed my thesis prospectus and Daniel with whom I worked together to calibrate the camera and also collected some test results. Last but not least, I like to thank United States Department of Defense (DoD), Air Force Institute of Technology (AFIT) and my sponsor, the Ministry of Defence (MINDEF) Singapore, Republic of Singapore Air Force (RSAF) and Air Logistics Department (ALD) for making my studies and thus this research at AFIT possible.

Kwee Guan Eng

Table of Contents

List of Figures

Page

List of Tables

Page

List of Abbreviations

INTELLIGENT BEHAVIORAL ACTION AIDING FOR IMPROVED AUTONOMOUS IMAGE NAVIGATION

I. Introduction

Navigation is the determination of a system's position, orientation and movements, and computing the maneuvers required to continue to the next location in the system's mission [1][2]. Without continuous precise navigation, deduced reckoning navigating systems do not have accurate information of its position and thus produce erroneous guidance information that fail to bring the system to the next intended waypoint. Therefore precise navigation is important to an autonomous navigating system.

A common navigation solution uses the Global Navigation Satellite Systems (GNSS) [3][4]. In situations where GNSS is not available (e.g. indoors, jamming etc), alternate navigation system such as image (or vision) navigation using stereo cameras [3][4] has been proposed to provide visual odometry information of the vehicle, similar to what had been extensively studied in the computer vision community [3].

However, image navigation is susceptible to large errors or loss of functionality in areas with few image features (e.g. featureless wall) [5]. Although much research has been carried out on image navigation (Section 1.2), none have specifically quantified the effects of feature locations on egomotion, and actively position the robot to place features at "sweet spots" in the images to improve image navigation accuracy. This research determines if robot actions can aid image navigation, improving its accuracy especially in areas with few image features.

1.1 Problem definition

Vision or image navigation uses visual identification of salient landmarks to guide the robot towards its goal [6]. Egomotion is a critical process in image navigation, where landmarks are tracked in an image sequence and the change in camera positions (hence

the robot position) is determined for each image frame by estimating the relative movement of the tracked landmarks in the camera frame of reference [7]. However, egomotion's reliance on having sufficient landmarks for tracking affects egomotion's accuracy (or use) in areas with few image tracking points [5]. In addition, if landmarks are not well distributed in the image, it is also difficult to reliably estimate the robot's position, thus creating egomotion errors [8]. An example of an area with few landmarks that are also sub-optimally distributed is a long corridor with plain (featureless) walls. Figure 1 shows a long corridor with landmarks marked with green crosses. Observe that the wall on the right is devoid of landmarks. If the left wall is also featureless, (which is not uncommon), there are too few usable landmarks along the corridor. Due to the effect of vanishing point geometry for a long corridor, it is common to observe that landmarks are typically clustered at the end of the corridor, appearing at the center of the image.

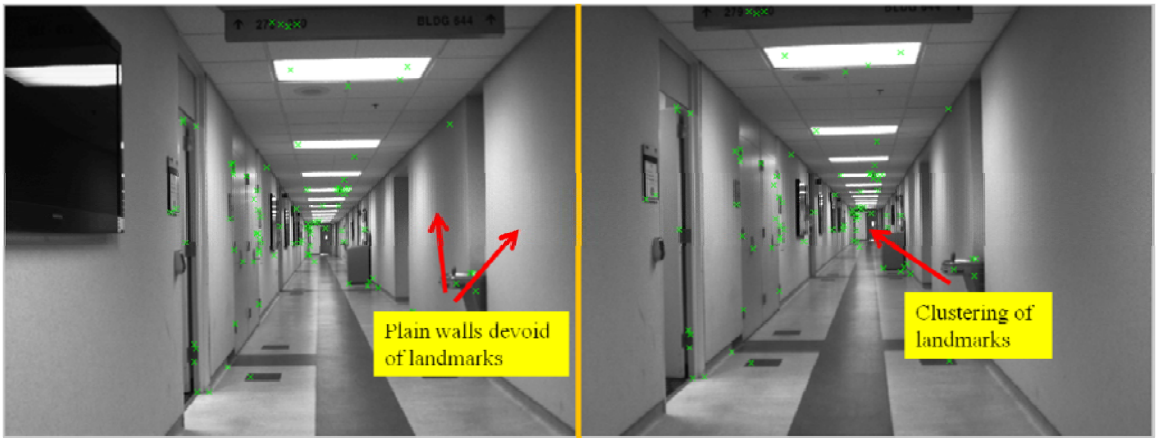

Figure 1: Left and right camera images showing identified landmarks (green crosses) along a long corridor.

To illustrate the egomotion errors when a robot navigates in a corridor environment, Figure 2 shows an egomotion path when a robot navigates along the corridors of the Air Force Institute of Technology (AFIT). It shows a plot of the egomotion path superimposed on the truth path taken by the robot [9]. The deviation of the egomotion path from the ground truth path illustrates the egomotion errors that occur when robots navigate along a long corridor.

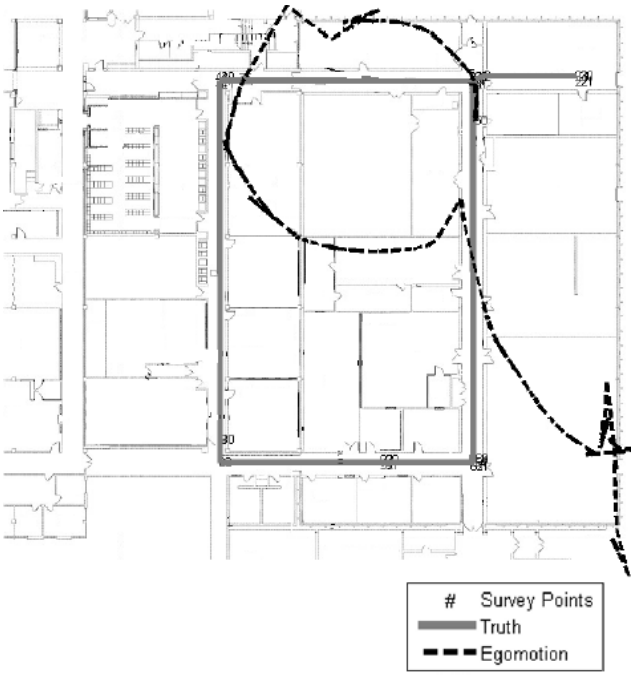

Figure 2: Robot egomotion path, together with the truth path. Adapted Figure [9].

1.2 Existing efforts to improve image navigation accuracy

There exist many proposals to reduce egomotion errors. Erroneous feature and landmark matching is one of the causes of egomotion error. To minimize this matching error, Olson, et al. [10] proposes limiting the search space for the corresponding feature / landmark pair only within the predicted feature / landmark location based on the robot motion. However other sensors (e.g. odometry, Inertial Measurement Unit) are required to measure the robot movement and estimate the projection of the landmark. However, this research aims to improve the accuracy of image-only navigation.

It is identified that optimal selection of landmarks allows the position of a robot to be determined through triangulation with less uncertainty [11] compared to using any landmarks. In the same research, it is also mentioned that if a robot's steerable camera pans and identifies optimally placed landmarks at each step of the journey, the uncertainty of a robot position is reduced by an order of magnitude. Olson, et al. [10] also suggests that egomotion error is reduced by selecting landmarks. However, no general

model exists that describes the relationship between landmark position and egomotion accuracy. In addition, landmarks availability is limited and it is not always practical to limit or choose which landmarks to use.

Bryson & Sukkarieh [12] show that implementing a fixed 'S' or 'Orbit' flight profile (for an UAV) increases the view of landmarks and improves image navigation heading accuracy. However, Bryson & Sukkarieh's proposal lacks evaluation capability and uses landmarks opportunely. Therefore, heading accuracy improvement is not optimal.

Active loop closing, route planning and landmarks relook which increase overall path and mapping accuracy have also been studied [13]. However, these techniques do not provide an immediate solution to egomotion errors for each robot step, but rely on having the robot return to already explored areas to relook at registered landmarks. From an operational point of view, it is sometimes impractical.

There are proposals that circumvent the fundamental image navigation problems by integrating image navigation with other navigation sensors such as inertial sensors [14] and odometry [9]. However, these approaches do not address the fundamental issues with image navigation, but work around them.

1.3 Research hypotheses and goal

It is hypothesized that the position of a landmark relative to the robot position and orientation (pose) affects egomotion accuracy when this landmark is used for egomotion calculation. Certain landmark positions give more accurate and consistent egomotion results compared to other positions. Therefore, active positioning of the robot to place landmarks at "superior" positions reduces egomotion errors and increase consistency.

The goal of this research is to deliver a robot behavioral algorithm that implements an intelligent action decision engine that reduces egomotion errors. It is also desired that action aiding increases the usability and reduces egomotion errors in areas with few features. The action aiding engine should be easily implemented on most existing image navigation solutions.

1.4 Potential applications

Since this research's proposed egomotion error reduction technique is based on robot action aiding, it is applicable to all existing image navigation solutions without major modifications to the existing egomotion engines. A lightweight and generally standalone algorithm that examines landmark locations and determines the action vector for the next robot step that is most beneficial for egomotion accuracy is developed. With increased egomotion accuracy, it could become feasible to implement standalone image navigation in small, less complex, low powered navigation systems and applications such as small robots on which space and power are limited. If used with other sensors, the overall performance of the navigation package is expected to be improved.

1.5 Research approach

This research first establishes the relationship between landmark positions and their effects on egomotion accuracy. A landmark location egomotion error model is then developed. Based on the observed relationship, three action aiding techniques are proposed: (1) non evaluative action aiding, (2) evaluative action aiding with physical scans before each step, and (3) evaluative action aiding with landmark tracking. A proof of concept is first performed in MATLAB simulation where relative performances are also compared (using a journey coefficient of variation developed in this research). Thereafter, the action aiding techniques are coded into C/C++ language and implemented on a Mobile Robots, Inc. Powerbot robot for testing in a real environment. The robot is routed along a truth path assisted by each of the action aiding techniques. For each test

run, the robot egomotion path is determined through post processing of collected images and the egomotion error distance (|truth stop position – egomotion perceive stop position|) is determined. With repeated tests, the mean egomotion error distance for each action aiding technique is derived. The research is considered a success if the average egomotion error distance with action aiding is smaller than without action aiding. Note that relative egomotion error distances between different action aiding techniques are used to assess performances in this research. Absolute egomotion error distances are not emphasized in this research as various implementation simplifications affect the absolute values.

Simulation results show that egomotion with no aiding has the highest journey coefficient of variation, followed by non evaluative action aiding, and evaluative action aiding has the lowest journey coefficient of variation. Actual testing on the robot in a controlled indoor environment validates simulation results. No aiding produces the largest average egomotion error distance, followed by non evaluative action aiding and evaluative action aiding has the smallest average egomotion error distance.

1.6 Organization of thesis

The following chapter presents relevant background knowledge essential for this research. Chapter 3 presents the concept and preparatory analysis leading to the development of the proposed action aiding techniques. The chapter firsts analyzes the effects of landmark locations on egomotion accuracy. Various action aiding techniques are then proposed together with an initial analysis. In chapter 4, a proof of concept and comparison of the relative performances between the various action aiding techniques are carried out. Action aiding techniques are then implemented on a robot for physical testing and the results are presented with detailed analysis. Based on the insights gained, chapter 5 proposes potential future works.

II. Literature Review

Measurement variations (i.e. stochastic measurements) that cause egomotion errors can be represented using probabilities. Therefore, general knowledge of measurement probabilities, distribution (e.g. Gaussian), statistics (e.g. mean, covariance, coefficient of variation), weighted average and sum of random variables are first introduced. Since this research addresses egomotion errors, it is necessary to understand how egomotion is determined; how image features are identified, landmarks and their locations determined through epipolar geometry calculation, and finally how egomotion is calculated. Causes of landmark location measurement errors in imaging systems are next discussed. This discussion forms the foundation for the relationship between landmark locations and egomotion accuracy that is discussed in the next chapter.

2.1 Basic probabilities and statistics in measurements

In the real world, most systems and measurements include some chance for errors and are considered stochastic [15]. When these stochastic measurements are used to calculate egomotion, egomotion values are also stochastic. A landmark location measurement and subsequent calculation of the robot's position and orientation (pose) and egomotion are estimates as true values are not known for certain. In this research, it is assumed that systems are linear (which is an acceptable assumption when operating within the operating region), noise is white (i.e. not time correlated) and Gaussian distributed. Therefore, linear operations can be performed on measurements while the distribution remains Gaussian. A Gaussian distribution probability density function [16] is mathematically described in Equation 1.

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{1}{2\sigma^2}(x-\mu_x)^2\right] \quad \textbf{(1)}$$

where:

X = random variable.
x = denotes the various values that is taken on by the random variable X.
μ_x is the mean of the various values of the random variable X.
σ^2 the variance of various values of the random variable X.

Encapsulated within a Gaussian distribution is the mean and variance (or standard deviation) information. The mean (μ_x) or expected value is the "best guess" of a Gaussian distributed estimate. It is also the value of the random variable where the probability of occurrence is the highest. This statistic is utilized when multiple measurements of the same nature are available and it is desired to obtain a single "best guess" value.

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \textbf{(2)}$$

where:

μ_x = mean of the random variable.
x_i = individual random variable (measurements).
N = number of random variables (measurements).

The variance (σ^2), which is the measurement estimate's "spread", "confidence" or "certainty value", is defined as the second moment about the mean. Often, a small variation exists between each measurement even if all controllable conditions remain the same. This variation is captured in the variance (or standard deviation) and can be calculated as follows:

$$\begin{aligned} {\sigma_x}^2 &= E\left[\left(X - E(X)\right)^2\right] \\ &= E[X^2] - [E(X)]^2 \end{aligned} \quad \textbf{(3)}$$

where:

${\sigma_x}^2$= variance value of the RV.
$E[\,]$ = is the expectation operator.

A similar parameter that is also used to describe an estimate's certainty level is the coefficient of variation (CV) [17] which is the normalized standard deviation of an estimate:

$$CV = \frac{\sigma_x}{\mu_x} \quad \textbf{(4)}$$

This research uses CV instead of covariance to describe a measurement's uncertainty level because the inverse of Signal to Noise Ratio is CV and this relationship is needed to develop the landmark location egomotion error model. Since conceptually CV and covariance represent the same information, they are sometimes used interchangeably in this research.

2.2 Combining stochastic measurements

When the robot observes the environment through the camera system, the image navigation system identifies multiple landmarks (depending on the environment) in each frame. Each of these landmarks can be used to independently determine the robot's egomotion. However, variations (and inaccuracies) in landmark measurements cause each landmark (from the same frame) to produce a different egomotion value. This thesis will also show that these landmarks do not provide the same level of egomotion accuracy when they are used to calculate egomotion. Hence, each landmark will be assigned a weighting which is the CV value. To arrive at a single egomotion estimate for an image frame, a weight average approach is used. In a typical robot journey, multiple image frames are collected. To determine the total robot displacement and the confidence level (CV) of this value for the journey, egomotion information from each frame is added using the concept of Sum of Variables [18].

2.2.1 Weighted average

In this research, the egomotion value calculated from each landmark is modeled as a Gaussian distributed measurement. Each landmark is pre-allocated with a "weighting"

which commensurate with the landmark's position effects on egomotion errors (when a landmark from that position is used to calculate egomotion). To combine multiple egomotion measurements within the same image frame in order to obtain a best guess of the egomotion value for the frame, weighted averaging is used. Equations 5 and 6 are respectively used to obtain the weighted average of the egomotion value and the certainty level (CV) of this egomotion value for a frame.

$$\mu_T = \frac{CV_2}{CV_1 + CV_2} z_1 + \frac{CV_1}{CV_1 + CV_2} z_2 \quad \textbf{(5)}$$

$$CV_T = \frac{CV_1 CV_2}{CV_1 + CV_2} \quad \textbf{(6)}$$

where:

CV_1 and CV_2= coefficient of variation associated with landmarks 1 and 2 respectively.
z_1 and z_2 = measurements associated with landmarks 1 and 2 respectively.
μ_T and CV_T = the combined mean and variance from the various measurements.

The weighted averaging equations are repeated for multiple egomotion measurements within the frame. The characteristics of the formula significantly influence the concept behind the research's proposed solutions. The combined uncertainty is never greater than the smallest uncertainty value from any single measurement. Having more landmarks regardless of their individual measurement uncertainty will not degrade the overall egomotion estimate (i.e. less uncertain of an estimate). Therefore, selection of landmarks in optimum positions as proposed by Olson, et al. [10] is not implemented in this research. This research uses all available landmarks for egomotion calculation.

2.2.2 Sum of random variables

Each egomotion step (dx, dy) is calculated independently. In a journey of N number of egomotion steps, the uncertainty associated with each step's measurement accumulates [19]. To obtain a single variance value that describes the uncertainty in a journey egomotion estimate, the Sum of Random Variable [18] technique is used to sum

each step's CV to form a single journey egomotion CV. This value is used to compare the relative performances (in simulation) of different action aiding techniques proposed in this research. Let two independent random variables (which can represent any two egomotion step estimate) be represented by X and Y with respective variances denoted by $Var(X)$ and $Var(Y)$. Given that another random variable Z is related to X and Y such that $Z = X + Y$. The variance of Z can be determined through Equation 7. When there are more than 2 variables (e.g. multiple landmarks each contributing an egomotion value for the frame), the equation is applied iteratively.

$$Var(Z) = Var(X) + Var(Y) \quad \textbf{(7)}$$

2.3 Image navigation

The preceding sections highlighted that all measurements are random variables which cause egomotion errors. It was also shown how these random variables are combined (for each frame) and added (from multiple frames) to obtain the egomotion step and journey value estimates. This section presents key principles of image navigation, specifically how robot egomotion is determined. The processes are broadly classified as (1) identifying image features (2) identifying landmarks, (3) determining landmark positions relative to the robot, and (4) establishing robot movements.

2.3.1 Features vs landmarks

This document refers to features as the image tracking points identified by Scale-Invariant Feature Transform (SIFT) [20]. When corresponding tracking points are matched between the left and right images, these are referred to as landmarks. Landmarks are suitable to be tracked for egomotion calculation.

2.3.2 Identifying image features - Scale Invariant Feature Transform (SIFT)

When a robot moves, image features scale, rotate, translates and viewpoint angle changes under varying illumination conditions. Only features that are recognizable through these manipulations are suitable for tracking. SIFT is a technique developed by Lowe [20] that is used in this research to identify image features invariant to scaling, rotation and translation, and partially invariant to illumination and viewpoint angle changes. As this research's tests are carried out in a controlled indoor environment with constant illumination, illumination variation effects are assumed negligible. Bebis [21] shows that as viewpoint angle difference increases beyond 30°, the probability for the same feature to be recognized reduces below 80% (Figure 3). To minimize erroneous landmark matching when landmark tracking is implemented in this research, landmarks with viewpoint angle changes greater than 20° are dropped.

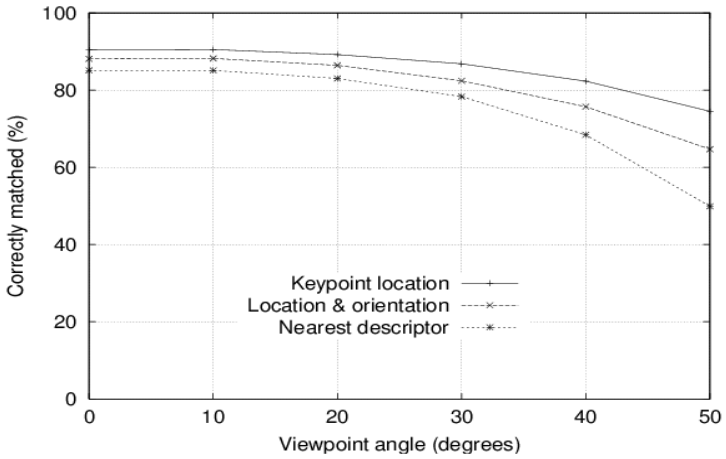

Figure 3: Probability for correct descriptors matching as viewpoint angle to the same feature changes [21].

SIFT identifies image features that have the characteristics described above by first smoothing (blurring) an original image using a Gaussian kernel and subtracting it from the original image to produce a difference image. This smoothed image is then sub-sampled to produce the next image scale and difference image. This process is repeated until either the image is too small for detection, or for a fixed number of scales (e.g. 16 scales). Points that are identifiable through the different scales are identified as SIFT features and the pixel locations of these points are found through the detection of maxima

and minima points through the different scales. Descriptors that uniquely describe these features are then formed [22]. From this, it is clear that both image resolution and sensor noise affect how accurately a feature location can be determined as resolution and noise shifts where the maxima and minima points are located. Effects of noise and resolution on feature measurement and egomotion errors are farther discussed in Section 2.4.1.

2.3.3 Identifying landmarks - Stereo matching

Although it is possible to determine the distance of a feature from a single still image through monocular cues such as texture variations, gradients, defocus, color, and haze [23], distance measurement of stereo matched SIFT features are more accurate [22]. Henceforth, stereoscopic vision is implied for depth perception. To determine the distance of a landmark from the robot, the same feature must appear in both left and right images that are taken simultaneously. A process called "matching" identifies corresponding feature pair from both left and right images to form a landmark. Section 2.3.2 pointed out that descriptors uniquely describe each feature. Hence, features with identical or relatively similar descriptors can be considered the same. The process starts with the calculation of descriptor Euclidean distance between a left image feature with all features on the right image. The descriptors of all right image features are first pre-transposed.

$$des\_all\_T_{im2} = des\_all\_{}_{im2}{}' \quad \textbf{(8)}$$

where:

$des\_all\_T_{im2}$ = transposed descriptors of all right (second) image features.

The left image feature descriptor is then dot product with all right image transposed features descriptors, and sorted from the smallest to the largest value.

$$dotprods = des_{im1} * des\_all\_T_{im2} \quad \textbf{(9)}$$

$$[vals, indx] = sort(\text{acos}\,(dotprods)) \quad \textbf{(10)}$$

where:

des_{im1} = A feature descriptor from the first (left) image.
$des\_all\_T_{im2}$ = transposed descriptors of all features in second (right) image.
$dotprods$ = dot products between the feature descriptor in the first (left) image and the transposed descriptors of all the features in the second (right) image.
$vals$ = sorted Euclidean distances from smallest to the largest.
$indx$ = position number of each dot product value before sorting.

With the distances sorted, the closest right image's feature descriptor is a prime candidate to be declared a match. For added assurance, a set of criteria is enforced to allow only the right image feature with a descriptor that is similar (i.e. small values, by setting a threshold) and significantly closer than the next closest feature (through the setting of the distance ratio parameter) to be matched. The same process is repeated for all other left image features and a list containing only matched features for each frame is maintained. These matched features are now declared landmarks. Any image features not matched are ignored.

$$\begin{aligned} &if\ \left(vals(1) < threshold\ \&\&\ vals(1) < distRatio * vals(2)\right) \\ &\{ \\ &\quad declare\ a\ match \\ &\} \end{aligned} \quad \textbf{(11)}$$

2.3.4 Determine landmark position – Epipolar geometry

Having identified landmarks, it is now desired to determine these landmarks' physical positions relative to the camera. Knowledge of a feature pair pixel locations in the left and right image are required to determine the landmark's physical position from the camera. This research implements the differential epipolar geometry technique described by Armangue [24] that is also implemented in the Multiple Integrated

Navigation Sensors (MINS) project by Weyers [9]. However, Izumi [25] presented the epipolar concept in a simpler manner, which makes other concepts subsequently presented in this research more easily understood. Hence, his explanation is presented here. Figure 4 illustrates the geometry involved in the calculation of the landmark horizontal location from the left camera. Without deriving, Equations 12, 13 and 14 respectively calculate a landmark's dx (distance of the landmark from the left camera axis), dy (depth/distance of the landmark from the camera imaging plane) and dz (height of landmark from the camera horizontal level plane) distances.

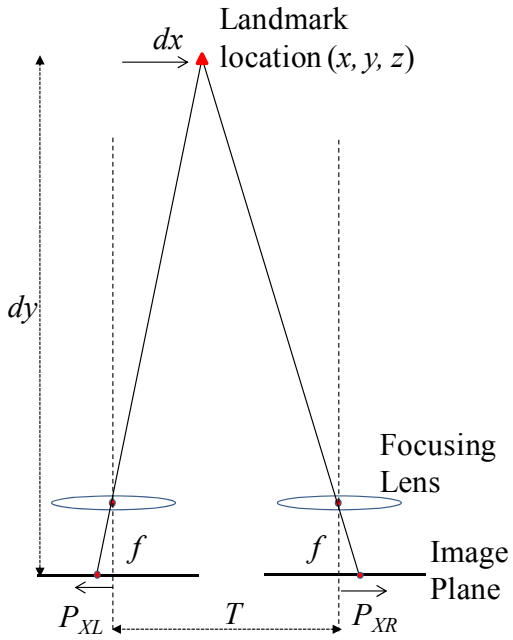

Figure 4: Geometry involved in Epipolar calculation (2D planar).

$$dx = \frac{P_{XL}T}{P_{XL} + P_{XR}} \tag{12}$$

$$dy = \frac{fT}{P_{XL} + P_{XR}} \tag{13}$$

$$dz = \frac{P_{YL}T}{P_{XL} + P_{XR}} = \frac{P_{YR}T}{P_{XL} + P_{XR}} \quad \textbf{(14)}$$

where:

P_{XL}, P_{XR} = pixel position that the landmark projects on the x-axis of the left and right image plane respectively.
P_{YL}, P_{YR} = pixel position that the landmark projects on the y-axis of the left and right image plane respectively.
T = inter camera distance.
f = focal length of the lens.

2.3.5 Establishing movement of the robot - Egomotion

Assuming static landmarks, the movement of landmarks in the images when a robot moves can be solely attributed to robot movements. Equation 15 relates a landmark image position before (X') and after (X) a robot movement, where T is the robot translation and R the robot rotation. Weighted Least Squares Minimization, Maximum-Likelihood Estimation [10], Iterative Closest Point [19] are some techniques that can be used iteratively to estimate T and R so that the solution to Equation 15 is valid. Other concepts based on "Motion Parallax", "Linear Subspace" are also studied [26]. Papers such as "Comparison of Approaches to Egomotion Computation" [26], "Vehicle Egomotion Estimation with Geometric Algebra" [27], "Robust Stereo Ego-motion for Long Distance Navigation" [10] are some papers available on the topic.

$$X' = RX + T \quad \textbf{(15)}$$

where:

X and X' = the same landmark position before and after the robot's movement.
T = translation.
R = rotation.

In this research, a full version of the egomotion algorithm is not implemented due to concerns of high processor computation load and various simplifications made. A simplified egomotion algorithm is thus developed that meets the requirements for relative performance comparison between the different robot action profiles (Section 4.4.7).

2.4 Causes of egomotion errors

In the ideal world where all measurements, equipment and models are perfect, there will be no egomotion errors. However, imperfections exist and this research raises camera sensor noise and resolution as two of the relevant causes for egomotion errors. They affect how accurately and consistently a static feature appears at a particular image pixel position. From Equations 12, 13 and 14, it is known that the precise image pixel positions (in the left and right images) of a feature is used to calculate the feature's physical location, which is farther used for egomotion calculation. Hence, errors in determining image pixel positions of a feature causes egomotion errors. Section 3.1 farther explores (experimentally) measurement variations.

2.4.1 Sensor noise

The camera sensor is made up of intensity detectors that are each a pixel. It is assumed that all detectors have white Gaussian noise of similar mean magnitude. However, the instantaneous noise magnitude on each detector is likely to be different. When a weak feature image centers between 2 pixels, noise level influences which pixel this feature is determined to be centered on, with the result swaying towards the pixel with the larger noise magnitude (Figure 5). Sensor noises are also recognized to affect other types of feature identification techniques such as Harris corner detection [27].

2.4.2 Resolution

The number and size of pixels on a sensor is finite. When the image of a feature is cast on the sensor, the graphic processor together with SIFT algorithm determine the nearest pixel the feature is centered on. This approximation creates quantization errors. Larger pixel size (low resolution) gives larger quantization errors (Figure 5).

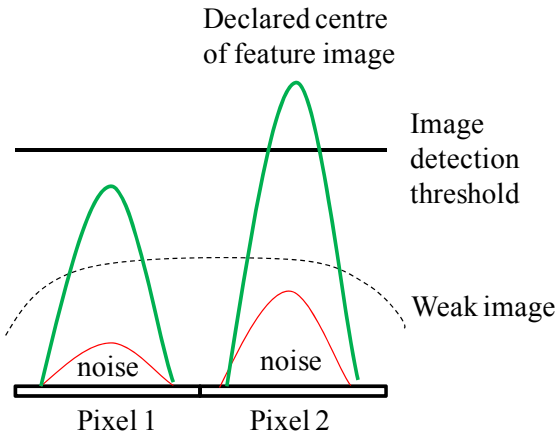

Figure 5: Noise biases the weak image center position towards the pixel with the larger noise.

In summary, when sensor noise is high and/or image resolution is low, image pixel position cannot be determined accurately and consistently. Landmark positions also cannot be calculated accurately, leading to increased egomotion errors.

2.5 Summary

This chapter discusses that in the real world, all measurements are stochastic (i.e. uncertainties exist). When there are multiple measurements, this research uses weighted averaging to combine and obtain the best measurement estimate, with a mean and CV value. It is also mentioned that the combined CV is smaller than the smallest single landmark uncertainty level, hence having more measurements reduce the combined uncertainty. The basic image navigation and egomotion concepts are also shown. This chapter also explains the causes of landmark measurement uncertainties in image navigation.

III. Methodology

Chapter 2 covered the knowledge required for this research. This chapter introduces the thought processes behind the development of robot action aiding techniques that reduce egomotion errors. This chapter first demonstrates that measurement variations and errors (e.g. due to sensor noise and resolution limitation) exist, and causes egomotion errors. The effects of landmark locations on both measurement and egomotion errors are studied next. A landmark location egomotion error model and equation, which describe how landmark locations and egomotion error are related are developed next. The error model shows the areas landmarks should be located at (relative to the robot) that provide the least egomotion errors (when they are used for egomotion calculation). Based on this, three action aiding techniques are developed: (1) non evaluative action aiding, (2) evaluative action aiding with physical scans before each step, and (3) evaluative action aiding with landmark tracking. In non evaluative action aiding, the robot moves in a "zig zag" profile down a corridor, reducing distances to landmarks and therefore, reducing egomotion error. Evaluative action aiding technique with physical scans before each step requires the robot to scan the environment at every step to determine the angle that contains landmarks which combined, give the least egomotion uncertainty contribution. Evaluative action aiding with landmark tracking tracks landmarks so that physical scans at every step are not required. This chapter ends by showing how the algorithm of an action aiding module is developed, specifically the various main functions and how data flows between them. Simulation and physical testing of the various action aiding techniques are presented in Chapter 4.

3.1 Existence of landmark measurement variation and error

In Section 2.4, it was theorized that sensor noise and quantization errors cause feature image pixel localization errors which lead to landmark location errors and egomotion errors. This section shows experimentally that landmark measurement variation and error do exists. The next section shows that egomotion error exists.

To demonstrate that landmark measurement variation and error exist, it is sufficient to show that even when all external controllable conditions stay the same (e.g. static robot and physical features, constant environment illumination etc), the measurement of an image landmark (e.g. distance, horizontal position and height) varies with each image frame. The experiment is conducted along a long corridor and the robot is stationary while 20 image frames of the static environment are collected. Five landmarks that appear through all frames are identified and their epipolar locations from each frame calculated. Figure 6 shows an image with all SIFT features while Figure 7 shows the five landmarks chosen for analysis. Visual observation shows that the same feature does not appear consistently at the same pixel location through all 20 image frames. Hence, when these feature pixel locations are used in epipolar calculations, variation in landmark locations occur. Table 1 summarizes the five chosen landmarks mean distance, horizontal position and height information. The standard deviation is also shown. The experiment shows the existence of measurement variations and errors.

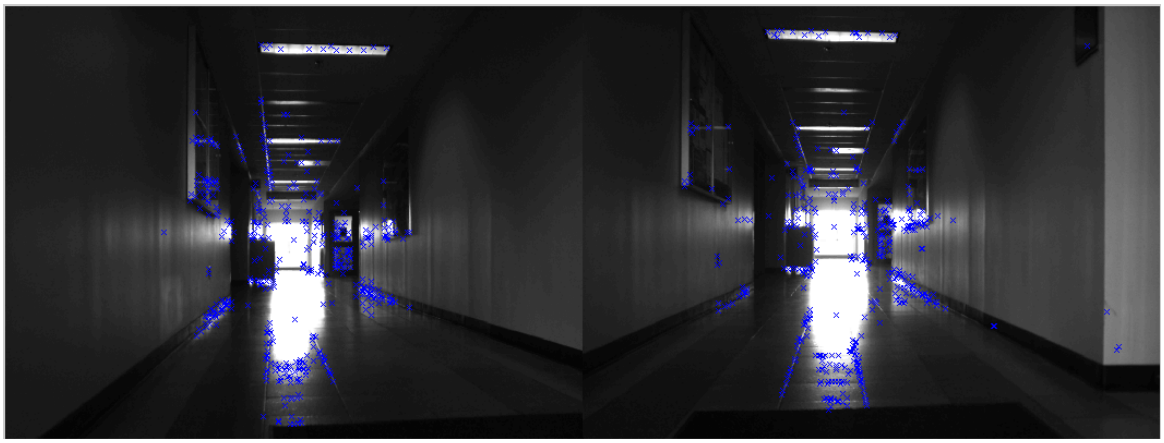

Figure 6: Example of a left and right camera image, shown with the features identified by SIFT (marked with a blue 'x' cross).

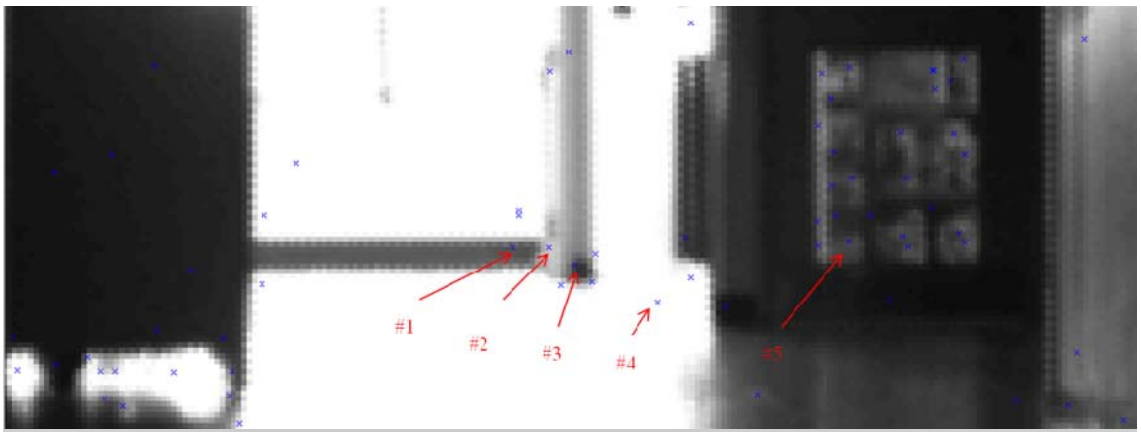

Figure 7: Landmarks that exists through all frames were chosen to tabulate their epipolar locations for each frame (zoomed in view).

Table 1: Mean and variance of the height, horizontal distance and depth for each landmark.

		Height (m)	Horizontal (m)	Depth (m)
Landmark 1	**Mean**	-0.76	-1.68	50.14
	Std Dev	9.7E-03	1.3E-02	5.0E-01
Landmark 2	**Mean**	-0.70	-1.26	45.49
	Std Dev	3.6E-03	8.6E-03	2.3E-01
Landmark 3	**Mean**	-0.83	-1.07	45.77
	Std Dev	3.2E-03	5.2E-03	1.6E-01
Landmark 4	**Mean**	-0.81	-0.38	34.90
	Std Dev	3.5E-03	1.9E-03	5.0E-02
Landmark 5	**Mean**	-0.48	0.75	34.25
	Std Dev	4.1E-03	1.1E-03	7.7E-02

Note: due to the orientation of the camera frame, negative height points upwards from the frame's origin

3.2 Effect of image landmark measurement errors on egomotion

This section shows the existence of egomotion errors caused by landmark measurement errors. To show the variation and errors in egomotion, each landmark in the same frame is independently used to compute egomotion. Since egomotion is calculated across one image frame, all landmarks should produce the same egomotion value. If however the landmarks produce different egomotion values, it is evident that egomotion errors occurred due to landmark measurement and localization errors. Table 2 shows the

egomotion values calculated from each landmark in a single image frame from images collected for Section 3.1. As observed, each landmark (from the same frame) produces a different egomotion value. Hence, it is seen that image landmark measurement variation and errors causes egomotion errors.

Table 2: Egomotion value for one image frame step from various landmarks.

Landmark #	Egomotion value (m)
1	0.18
2	0.13
3	0.12
4	0.10
5	0.08
6	0.56
7	0.50
8	-7.90
9	-4.41
10	0.10
11	0.51
12	0.44
13	1.76

3.3 Landmark position egomotion error model

This research hypothesized that landmark positions affect egomotion accuracy. Highlighted in Section 1.2, Claus [11] suggests choosing landmarks to determine the position of a robot. In doing so, the robot position accuracy is improved. Olson, et al. [10] also suggests that egomotion error is reduced through optimal landmark selection. This section first studies the effects of landmark positions on landmark measurement errors. Thereafter, the effects of landmark positions on egomotion errors are examined. Finally, the landmark position egomotion error model is developed.

3.3.1 Landmark position effects on measurement uncertainties

Table 1 shows that there is a relationship between landmark positions and landmark measurements. Illustrating using distance measurement, with increased landmark distance, distance measurement standard deviation increases. Figure 8 illustrates this observation. Interpreted, this means that travelling along a longer corridor gives greater landmark measurement uncertainties and errors (as most landmarks are farther away) compared to a shorter corridor that gives smaller landmark measurement uncertainties and errors (as landmarks are nearer). These observations have also been recognized by other researchers. Se [22] noted that with farther landmark distances, distance perception of these landmarks becomes less accurate.

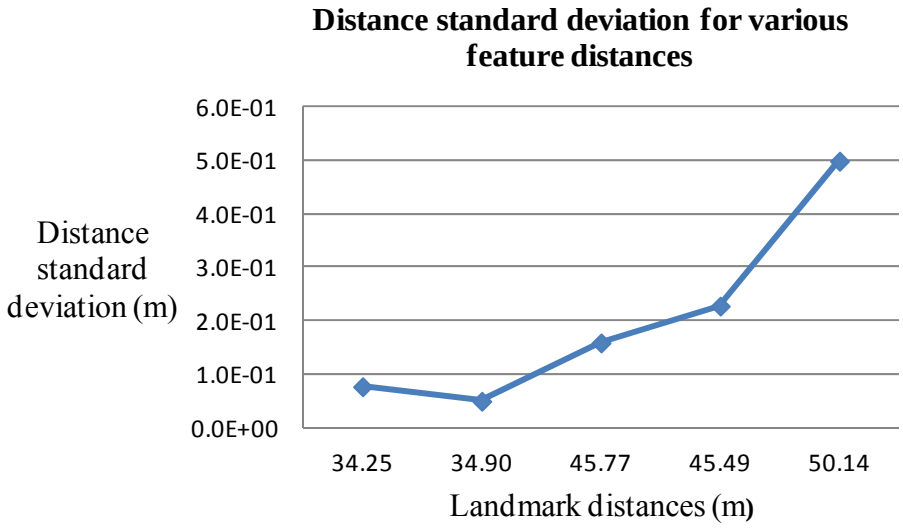

Figure 8: Relation between distance measurement standard deviation, and the distance of the landmark.

Table 1 also shows that landmark height and horizontal distance measurements standard deviations are significantly smaller compared to distance measurements standard deviation (Figure 9). This implies that there is significantly less uncertainty in planar measurements compared to distance measurements. When horizontal distance measurements are used for egomotion calculations, there is less uncertainty in the

calculation of angular motions compared to translational egomotion which are calculated from landmark distance measurements.

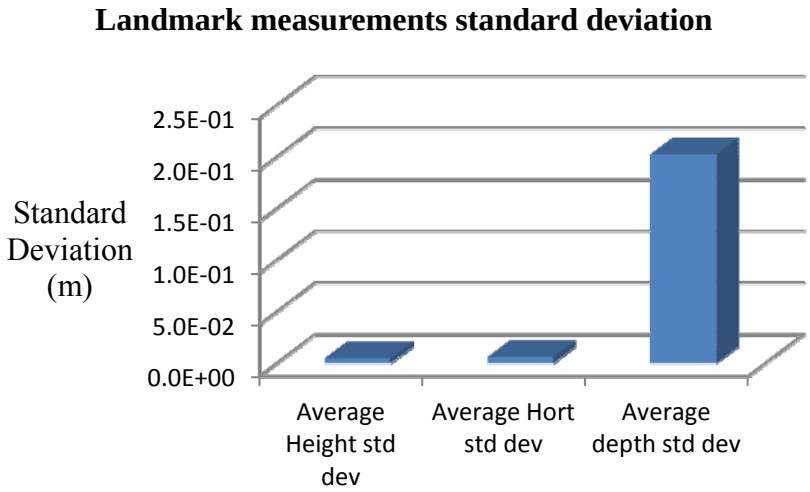

Figure 9: Difference in standard deviation for differing landmark measurements parameters.

To better represent landmark measurements standard deviation statistics, the same experiment (collecting images while the robot and environment are static, and determining the standard deviation of landmark measurements) was repeated to collect 100 image frames. Figures 10 and 11 show the landmark distance and horizontal measurement (respectively) standard deviation along one horizontal displacement plane for various landmark distances. Figure 10 shows that landmark distance measurement standard deviation increases with landmark distances, while Figure 11 shows that landmark horizontal displacement standard deviation does not differ as much with landmark distance. Again comparing Figures 10 and 11, it is noticed that landmark horizontal measurement standard deviation is about 7 times smaller than landmark distance measurement standard deviation.

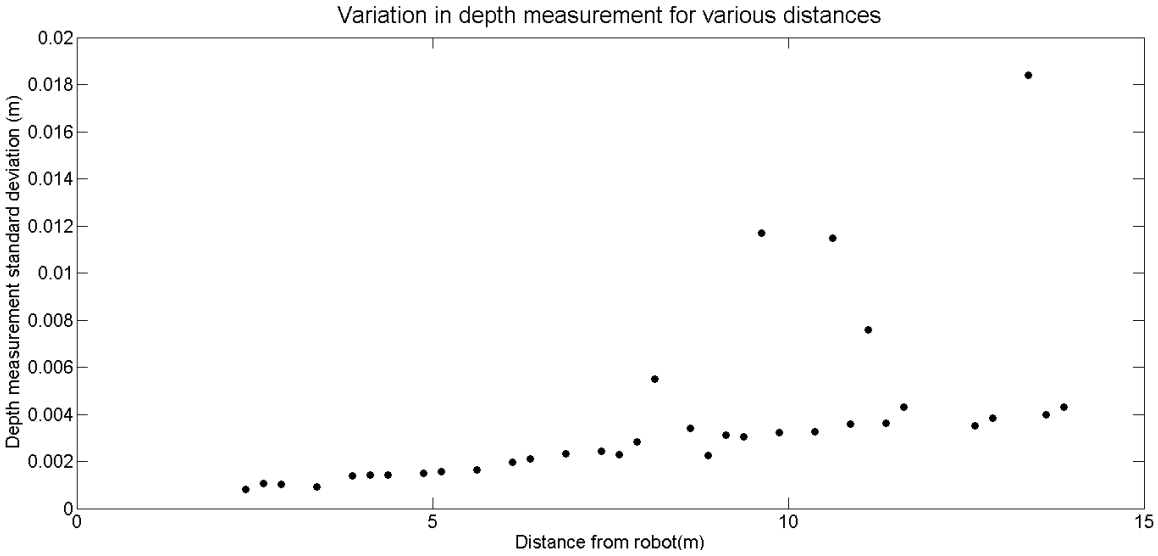

Figure 10: Standard deviation in depth measurement for landmarks at various distances.

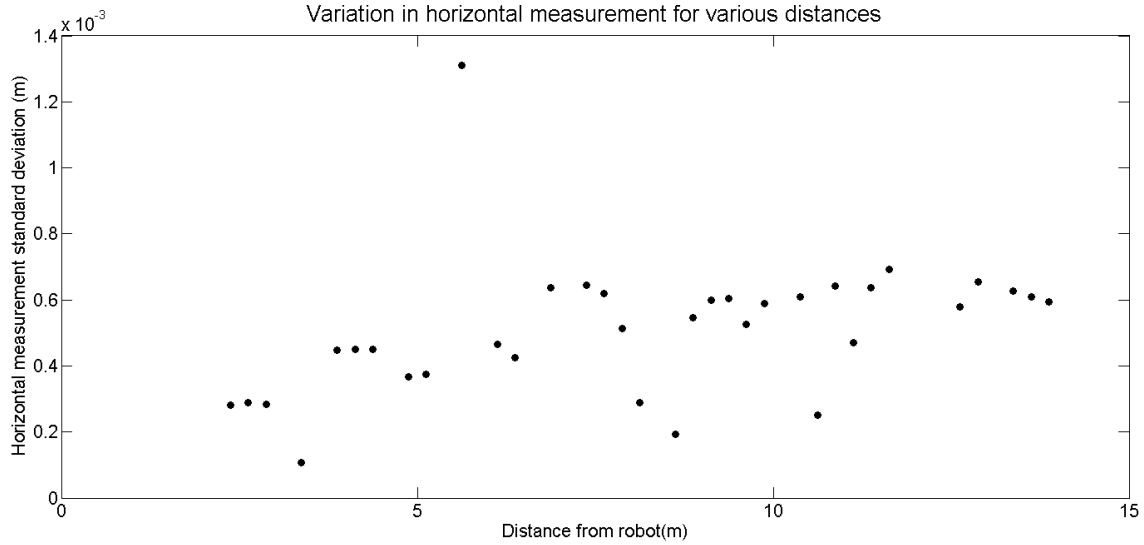

Figure 11: Variation in horizontal distance measurement for landmarks at various distances.

There is sufficient evidence to conclude that landmark distance measurements become increasingly uncertain with increase in landmark distances. Therefore, it is expected that translational egomotion becomes increasingly uncertain and erroneous with increased landmark distances. Since landmark horizontal measurements standard deviation is smaller, it is expected that angular egomotion has smaller errors compared to translational egomotion errors.

3.3.2 Landmark position effects on egomotion accuracy

To recap, this research hypothesized that landmark positions affect egomotion accuracy. The previous section demonstrated that landmark distances affect landmark measurement errors. This section examines if increased landmark distances translate to increased egomotion errors. Referring to Figure 4 (Section 2.3.4), changes in P_{XL} and P_{XR} is indicative of egomotion. Figure 12 illustrates the changes in P_L and P_R in response to different landmark distances. Both left and right configurations are identical except for the difference in landmark distance (y). δy represents the change in landmark position from the robot when the robot moves a step. δy is the same distance for both setups. By visual observation, notice that when the landmark is located farther (left configuration), δy induces a small change P_L and P_R. In contrast, when the landmark is closer (right configuration), the same δy cause a larger change in P_L and P_R. In summary, for the same robot distance moved (δy), the magnitude of P_L and P_R ($|\delta P_L + \delta P_R|$) changes depending on the location of the landmark. The farther the landmark, the smaller the $|\delta P_L + \delta P_R|$. Therefore, $|\delta P_L + \delta P_R|$ is analogous to egomotion calculation signal power (P_s). Assuming the camera sensor has the same mean white Gaussian noise for all pixel detectors, P_s is indicative of the egomotion calculation Signal to Noise Ratio (SNR). The inverse of SNR is CV [17]. Therefore, higher SNR gives smaller egomotion uncertainty (smaller CV) and greater egomotion accuracy, while lower SNR gives larger egomotion uncertainty (larger CV) and lesser egomotion accuracy. Table 3 summarizes the effects landmark distance has on egomotion CV and errors.

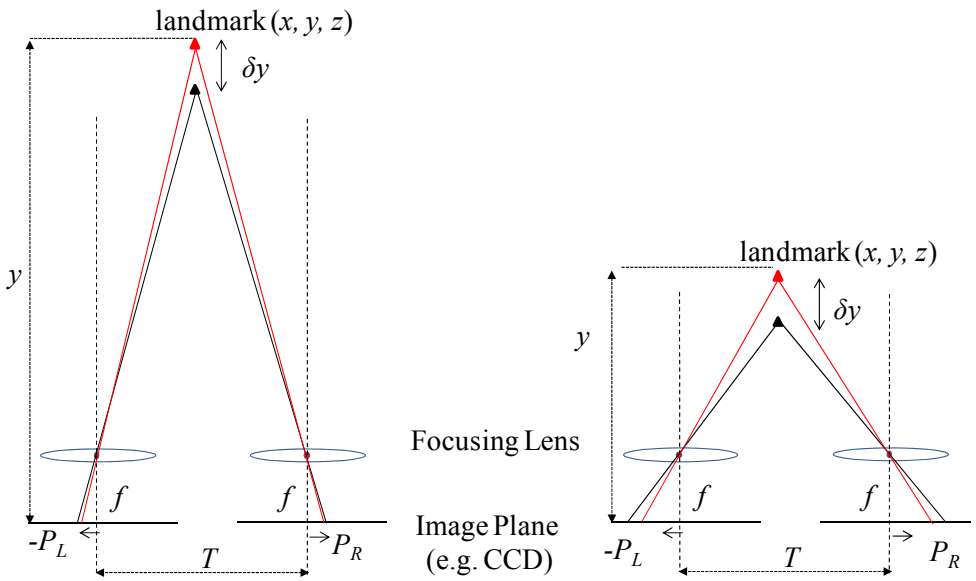

Figure 12: Different magnitude change in PL and PR for the same change in landmark position between frames (δy) for landmarks at different locations.

Table 3: Summarizing landmark distance effects on egomotion CV and error.

Landmark distance	$\lvert\delta P_L + \delta P_R\rvert$	P_s	SNR	Egomotion CV	Egomotion error
Far	Small	Small	Small	Large	Large
Near	Large	Large	Large	Small	Small

An experiment was carried out to farther determine the effects of landmark positions on egomotion accuracy. The robot collects translation images while it is driven manually at a constant speed (same distance moved between successive image frames) along the corridor. An algorithm was developed using MATLAB to post process the collected images to determine the egomotion value. For each successive image frame, landmarks are identified and locations determined. Egomotion is calculated from each landmark and grouped into various physical regions the landmark is located. The average egomotion value (if multiple landmarks are identified in that region) for each region is determined, which represents the estimated robot egomotion value if a landmark located at that position is used to determine egomotion. As it is difficult to obtain a large number of egomotion values within a single region, standard deviation statistics is not representative. Hence, the egomotion distance information is plotted across all the

regions, and the variation across regions is visually examined. Figures 13 and 14 show the egomotion values (distance advanced by the robot between each frame) determined by landmarks located at different regions from the robot. As the camera image capture rate is 2Hz and robot traveled at a constant 0.3m/sec, the correct egomotion distance is 0.15m. Since the image processing and egomotion algorithm used are simplified versions without advance data processing techniques such as outlier rejection, feature location consistency checking etc, outliers and negative distances are observed.

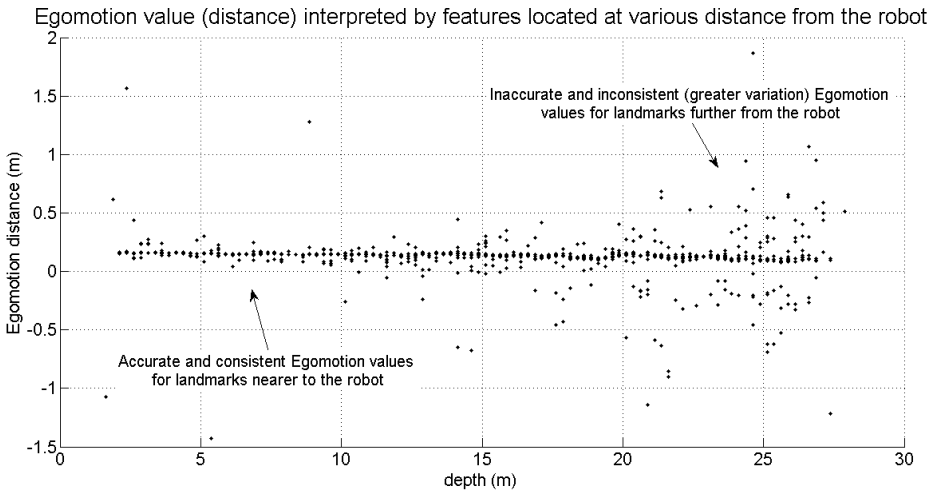

Figure 13: Egomotion value determined by landmarks located at different distances from the robot.

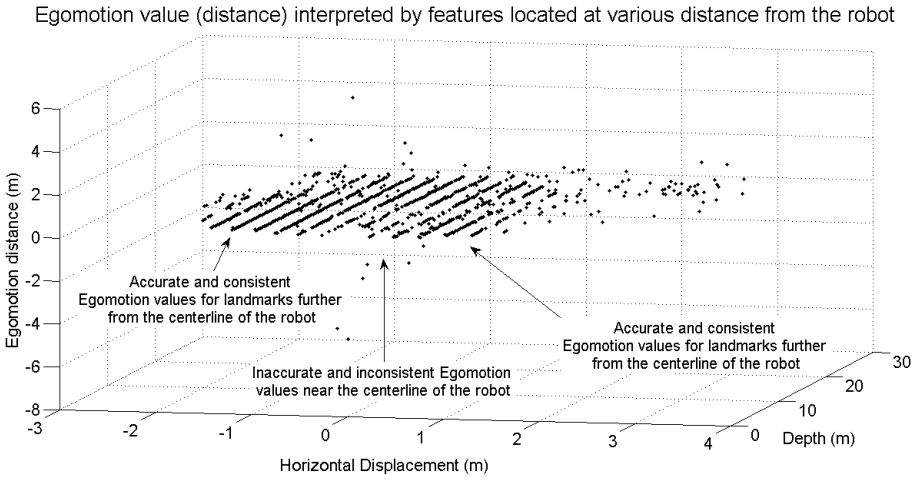

Figure 14: Egomotion value determined by features at different locations (3D view).

Figure 13 shows that landmarks near to the robot determine egomotion values close to 0.15m (truth value) and the variation of egomotion values is small. When the landmarks are located farther, the variation in egomotion values increases (larger spread of the plots) and the egomotion values are also farther from 0.15m (i.e. greater errors). The results imply that nearer landmarks give more accurate egomotion values and with greater consistency. Farther landmarks give more egomotion errors and greater uncertainties. Figure 14 shows that landmarks located along / near the robot centerline give more egomotion errors and variation while those nearer to the robot's Field Of View (FOV) edge are more consistent. This implies that if the robot is translating and landmarks directly in front of the robot are used to determine egomotion, more egomotion errors are expected. If landmarks to the side of the camera image (i.e. closer to the camera FOV edge) are used, egomotion results are more accurate and with less uncertainties.

3.3.3 Landmark position egomotion error model

To derive a mathematical model describing the relationship between landmark locations and egomotion error, the egomotion SNR ($|\delta P_L + \delta P_R|$) for robot movement of 1m (δy) is calculated for all landmark positions and the inverse value (CV) determined (Equations 16 to 23). Note that the various CV values are only used for relative comparison. The absolute value has no interpretation meaning in this research. The plot of egomotion CV for landmarks at different positions is the landmark position egomotion error model (Figures 15 and 16).

$$P_{L1} = f * \tan\left[tan^{-1}\left(\frac{xLm^{(g)} - xL^{(g)}}{yLm^{(g)} - yL^{(g)}}\right)\right] \quad \textbf{(16)}$$

$$P_{R1} = f * \tan\left[tan^{-1}\left(\frac{xR^{(g)} - xLm^{(g)}}{yLm^{(g)} - yR^{(g)}}\right)\right] \quad \textbf{(17)}$$

$$P_{L2} = f * \tan\left[tan^{-1}\left(\frac{xLm^{(g)} - xL^{(g)}}{yLm^{(g)} - step - yL^{(g)}}\right)\right] \quad \textbf{(18)}$$

$$P_{R2} = f * \tan\left[tan^{-1}\left(\frac{xR^{(g)} - xLm^{(g)}}{yLm^{(g)} - step - yR^{(g)}}\right)\right] \quad \textbf{(19)}$$

$$\delta P_L = abs(P_{L1} - P_{L2}) \quad \textbf{(20)}$$

$$\delta P_R = abs(P_{R1} - P_{R2}) \quad \textbf{(21)}$$

$$P_s = \delta P_L + \delta P_R \quad \textbf{(22)}$$

$$CV_{ego\_lm} = \frac{1}{P_s} \quad \textbf{(23)}$$

where:

P_{L1}, P_{R1} = Landmark image pixel position on the left and right camera sensor respectively.
P_{L2}, P_{R2} = Landmark image pixel position on the left and right camera sensor respectively, after robot movement of size "step".
$\delta P_L, \delta P_R$ = Landmark image pixel position change on the left and right camera sensor (respectively) after robot movement of size "step".
P_s = Signal strength for Egomotion calculation.
$xLm^{(g)}, yLm^{(g)}$ = Landmark x, y coordinates in global frame.
$xL^{(g)}, yL^{(g)}$ = Left camera sensor center point x, y coordinates in global frame.
$xR^{(g)}, yR^{(g)}$ = Right camera sensor center point x, y coordinates in global frame.

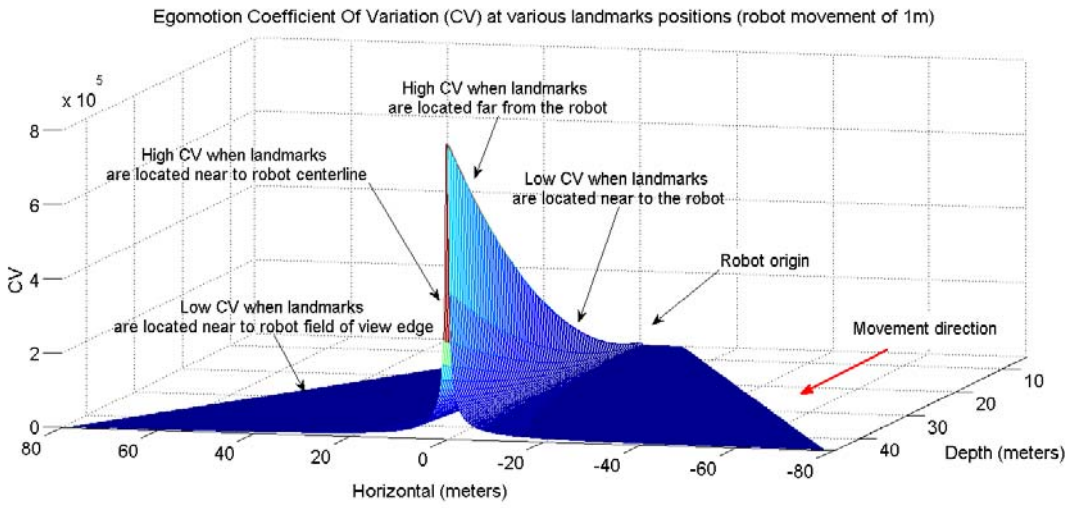

Figure 15: 3D egomotion error model - representation of the egomotion CV for various landmarks positions.

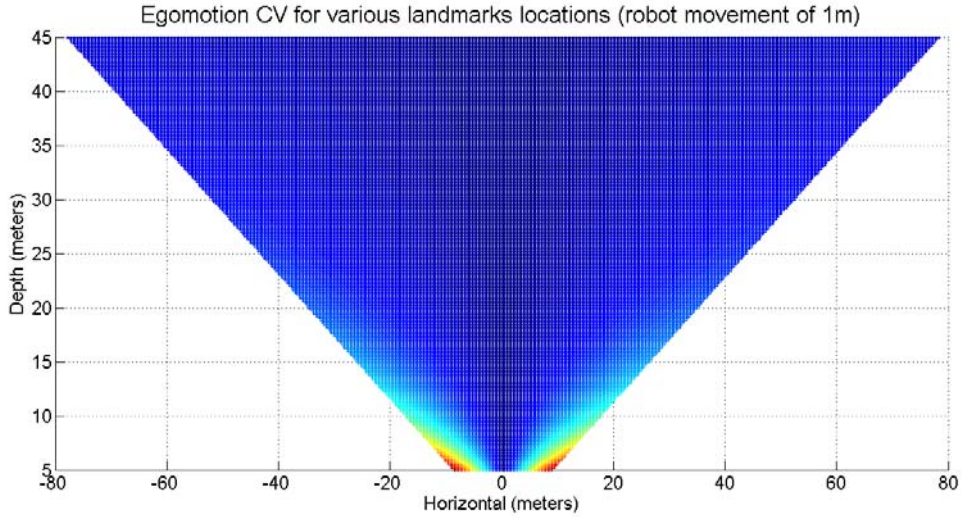

Figure 16: 2D egomotion error model - representation of the egomotion CV for various landmarks positions. Increasing shades of blue indicates increasing egomotion CV.

The landmark position egomotion error model can be described as a "3D exponential" curve shown in Figure 17 and mathematically expressed by Equation 24. When landmarks directly ahead (near the centerline) of the robots are used to calculate egomotion, egomotion errors are large. When landmarks located near to the robot's FOV are used to determine egomotion, egomotion errors are smaller (compared to egomotion

errors calculated from landmarks located near to the robot centerline). Landmarks nearer to the robot give more accurate egomotion values compared to landmarks that are located farther from the robot.

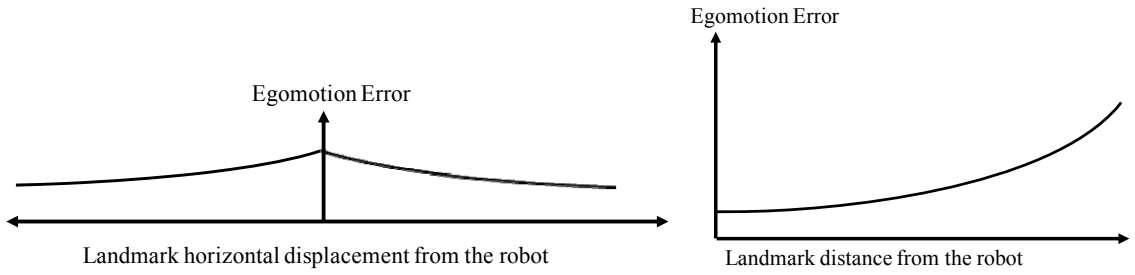

Figure 17: Notional landmark position egomotion error model.

The egomotion CV contribution values associated with each landmark location has extensive use in this research. Therefore, it is necessary to formulate a mathematical equation (Equation 24) that describes the landmark position egomotion error model, so that given a landmark position from the robot, the egomotion CV associated with this landmark position can be calculated. This equation should represent a "3D" exponential function. A growing exponential describes the increase in egomotion errors with distance (y). This value forms the initial amplitude of a second decaying exponential that describes the reduction in egomotion errors as a landmark gets increasingly farther displaced (x) from the center line of the robot. The parameters (α, β, γ) associated with the equation can be empirically tuned to fit the robot camera system setup and parameters. To reduce computation requirements for real time applications, the egomotion CV value associated with each landmark position are pre-computed in this research, and complied into a look-up table.

$$CV_{lm}(x, y) = (\alpha e^{\beta y}) e^{-\gamma |x|} \quad \textbf{(24)}$$

where:

$CV_{lm}(x, y)$ = Egomotion CV value associated with a landmark position from the robot.
x, y = Position of the landmark from the robot's pose.

α = Initial amplitude of the growing exponential. It is the egomotion CV value for a landmark directly in front of the robot.
β = Grow factor for the growing exponential. It describes the growth in egomotion errors with increasing landmark distances.
γ = Decay factor for the decaying exponential function. It describes the reduction in egomotion error as a landmark gets closer to the robot's FOV.
$\alpha e^{\beta y}$ = Initial amplitude of the decaying exponential at various landmark distances from the robot. It describes the egomotion CVs for landmarks on the centerline of the robot, at various distances.

3.4 Action aiding techniques

The key objective is to reduce egomotion error contribution from landmarks used for egomotion calculation. Observing the error model, qualitatively the landmarks should preferably be located near the robot and towards the robot's FOV edge. Quantitatively, the combined CV (using weighted averaging) from landmarks in the robot's FOV for each robot step should be minimal. These objectives can be achieved via repositioning of the robot pose to adjust the positions of landmarks in the robot's FOV. The robot should avoid travelling directly towards landmarks as landmarks along the robot centerline give large egomotion errors. The robot should maneuver itself to maximize the presence of landmarks near its FOV edges since landmarks at these locations give smaller egomotion errors. The robot should also position itself to reduce the distance to available landmarks as in general, nearer landmarks provide better egomotion accuracies. Since egomotion errors reduce with more measurements, the action aiding engine should also attempt to use all available landmarks.

The challenge for the action aiding technique is to find an optimum combination of the above-mentioned considerations given the environment and the robot pose, to obtain a robot action that produces an image with landmarks that combined, contributes the least possible (for that situation) egomotion errors. Three action aiding techniques are proposed: (1) non evaluative action aiding, (2) evaluative action aiding with physical scans before each step, and (3) evaluative action aiding with landmark tracking.

3.4.1 Non evaluative action aiding

Inspired by the studies conducted by Bryson and Sukkarieh [12], the robot does not travel in a straight line along a corridor. Instead, it travels in a "zig zag" (or "S") profile. Figure 18 illustrates the general path taken.

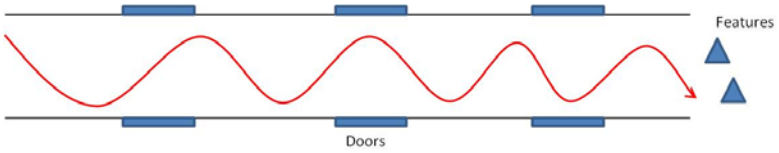

Figure 18: "Zig zag" (or "S") profile when traveling along a long corridor.

Concept

This is a qualitative technique where the main objective is to reduce the distances of landmarks seen by the robot. By "angling" the robot heading slightly towards the walls as the robot travels along the corridor, the image landmarks in the direct front view of the cameras are those along the walls and not those at the end of the corridor. The landmarks on the walls are closer compared to landmarks at the end of the corridor which the robot would have observed directly ahead if it travels in a straight line towards the end of the corridor. In addition landmarks at the end of the corridor now appear at robot's FOV edge most of the time which is an improvement compared to appearing directly ahead of the robot when the robot is not action aided. See Figures 19 and 20.

Figure 19: Illustrating the distance to landmarks observed by the robot if it faces straight down the corridor.

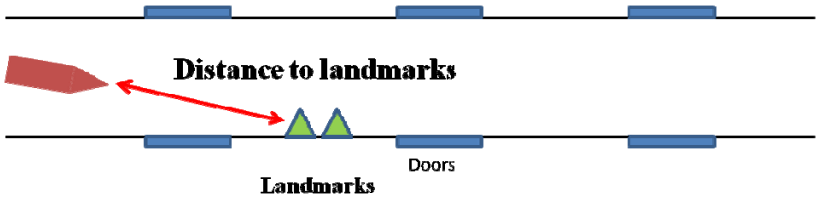

Figure 20: Illustrating the distance to landmark observed by the robot if it "angles" towards the wall.

Advantage

Nearer landmarks (along the walls) give smaller egomotion errors then farther landmarks (end of the corridor) when used for egomotion calculation. The landmarks at the end of the corridor now appear near to the robot's FOV edge and these landmarks also give less egomotion errors (compared to the egomotion errors the landmarks would have provided when they are in the robot's direct frontal FOV) when used for egomotion calculation. While more turns are required, it does not significantly increase egomotion errors since it was shown that landmark horizontal positions can be measurement accurately compared to distance measurements (hence more accurate angular egomotion compared to translation egomotion). Although taking more steps to reach the destination increases uncertainty, the increase in the number of steps (travel distance) is not significant since the angles taken by the robot is not large (narrow corridor). Improvements from tracking nearer landmarks on egomotion errors outweigh the degradation brought about by the slightly more steps (distance).

Short comings

The paths taken and turns (positions and angles) made are not based on evaluating landmarks locations. Landmarks used for egomotion tracking are opportunely selected based on the heading the robot happens to take at that instance. Hence while the average egomotion error reduces, the egomotion error standard deviation is large. During some runs, the robot faces a direction that places landmarks in the right positions while at other

runs, the landmarks are in poor positions. These cause the large variations in egomotion errors. In areas with large featureless walls, egomotion could fail.

3.4.2 Evaluative action aiding (physical scan)

This is a quantitative method where the algorithm analyzes landmark locations and finds the travel orientation with minimal egomotion error contribution from landmarks in the robot's FOV in each step.

Concept

Before each step, the robot physically rotates through the allowable limits (set so that the robot will travel down the corridor) to scan its surroundings. At each scan angle, available image landmarks are identified and egomotion CV contribution from each landmark determined (from the error model). Thereafter, the combined egomotion CV for the angle is predicted using weighted averaging. This process is repeated through all scan angles (Figure 21). The orientation that gives the lowest predicted egomotion CV is chosen. When the step is completed, this process is repeated, until the destination is reached.

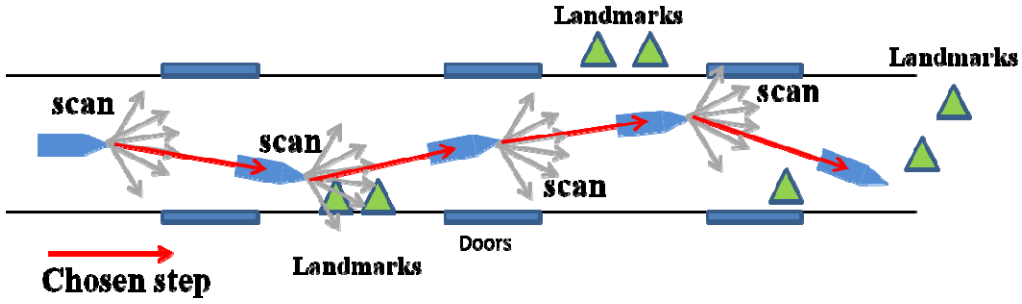

Figure 21: Evaluative action aiding technique with physical scans at each step.

Advantage

The robot has full awareness of the image landmark locations within its allowable travel limits. It has the capability to react to changing environments, and differing image landmark locations, determining the angle for each step that gives the least CV in egomotion calculation.

Short comings

The robot has to stop at every step for a thorough physical scan which is a lengthy process. Excessively rotating through large angles at each step introduces angular errors especially if odometry is used in robot movement measurements. These limitations are the key drivers for the development of the evaluative action aiding technique with landmark tracking.

3.4.3 Evaluative action aiding with landmark tracking

Evaluative action aiding with landmark tracking is a revision of the evaluative action aiding (physical scan) technique. The need to perform physical scanning before each step is removed through the addition of landmark tracking capability and maintaining a list of registered landmarks.

Concept

The addition of landmark tracking capability helps the algorithm "remember" where usable landmarks are located, even if they are not visible in the current robot FOV. At each step, the angles are algorithmically scanned, instead of physically. For each angle assessed, the algorithm determines which landmarks could be in view by analyzing the landmarks registered in the list. The combined egomotion CV (from multiple landmarks)

for each angle is then computed (using the same method described in Section 3.4.2) and the angle with the smallest egomotion CV is selected for the next robot step.

Advantage

Evaluative action aiding with landmark tracking has the capability to adapt to differing environment features distribution. With landmark tracking, the robot has good awareness of the usable landmarks that it can acquire even though the landmarks may not be directly observed in the present robot FOV. In addition, the robot does not need to physically scan all possible angles before each step which significantly reduces the process time.

Short comings

With possibility that there are undiscovered and unregistered landmarks at angles under consideration (which might have been discovered if physical scanning took place), the predicted egomotion CV value for the angle assessed might not represent the truth, affecting the robot orientation decisions.

3.5 Robot implementation

To provide a test control, it is necessary to determine the egomotion error when the robot is not action aided; hence the use of the term “no aiding” in this paper. Two action aiding techniques are tested using a robot in a controlled indoor environment: (1) non evaluative action aiding, and (2) evaluative action aiding with landmark tracking. Since evaluative action aiding with physical scans adds a considerable amount of process time which might not be suitable for actual deployment, it is not developed beyond simulation testing for physical testing. No algorithm is developed for “no action aiding” and non evaluative action aiding as the robot is driven manually during testing. For evaluative action aiding with landmark tracking, the robot has to navigate autonomously

with action aiding. Therefore, an evaluative action aiding (landmark tracking) algorithm is developed for implementation in the robot.

As the research uses different navigation frames of reference, the frames are discussed first in this section. The development of the algorithm for evaluative action aiding with landmark tracking is described next. The Unified Behavioral Framework is also introduced as it is beneficial to understand how the action aiding module integrates with the other robot operational systems and software.

3.5.1 Frames

The physical space (or the map) is fixed while the robot moves in it. Depending on how the camera is mounted on the robot platform, there can be separate frames for the camera and the robot. Therefore, three frames are introduced.

3.5.1.1 Global frame

This frame coordinatizes the physical space / map where the robot, landmarks, physical objects (e.g. walls, doors etc) are located. It is a fixed frame which does not rotate or translate regardless of robot movements, with its origin fixed at the position and orientation where the robot first starts navigating. For this research, it is assumed that the robot always starts from the same fixed position and orientation. For ease in development, the global frame origin is fixed at the bottom left corner of the navigation map. Variable x represents the horizontal axis and y is the depth axis (Figure 22). Since this research is in 2D, height information is ignored in most calculations. The robot and landmarks locations are represented, and most egomotion calculations (e.g. position and orientation) are carried out in this frame.

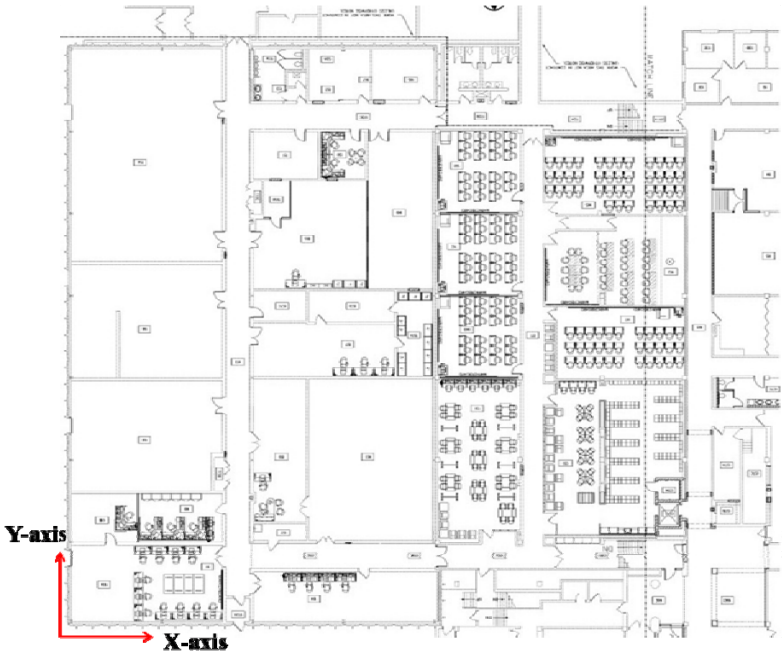

Figure 22: Global frame origin and axes.

3.5.1.2 Camera frame

The camera frame (Figure 23) coordinatizes the environment relative to the camera pose. The frame's origin is in line with the cameras' sensor plane, between the two cameras (stereoscopic two-camera system). It rotates and translates with the camera's movements.

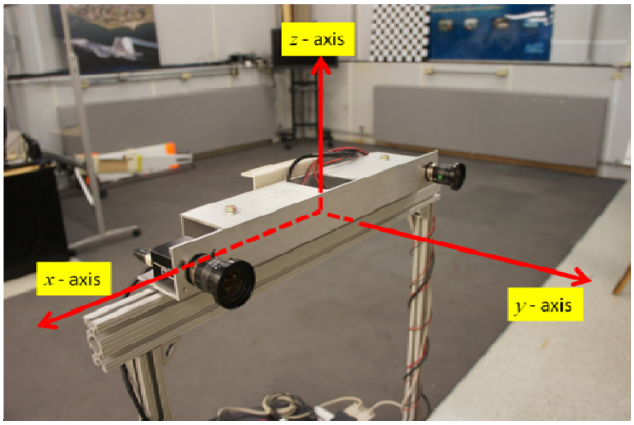

Figure 23: Camera frame.

3.5.1.3 Robot Frame

This frame which coordinatizes the robot's movements rotates and translates with the robot. The frame's origin lies on the intersection of the robot's rotation and translation axis (Figure 24). This frame is used when the desired robot movement determined from the action aiding engine is required to be translated to an actual robot movement.

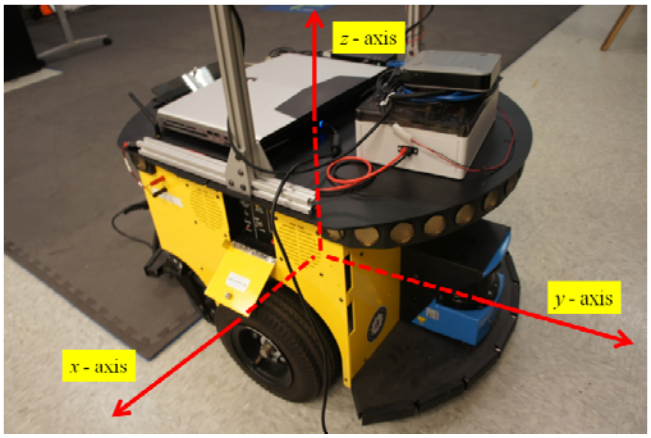

Figure 24: Robot frame.

3.5.1.4 Alignment of the camera frame with robot frame

If the camera and robot frames are not aligned, a Direction Cosine Matrix (DCM) is needed for the conversion. For implementation ease, the camera system is setup directly above the robot's wheel axle (Figure 25) at equal distance apart from the robot centerline (Figure 26). Since the robot is only capably of 2D movement, height is not important. Therefore this setup effectively aligns the camera and robot frames, and a DCM is not required. Due to inaccuracies in the installation of the cameras on the metal structure, and vibrations of the structure as the robot moves, the 2 frames are not accurately aligned and cause calculation errors. However, for the purpose of this research where the main objective is to compare the relative performances of the different action aiding techniques and not to obtain the absolute egomotion value, this small error is tolerable. In fact, simplifications made in the other parts of the system (e.g. egomotion algorithm) introduce more errors.

Figure 25: Cameras installed directly above the wheel axle.

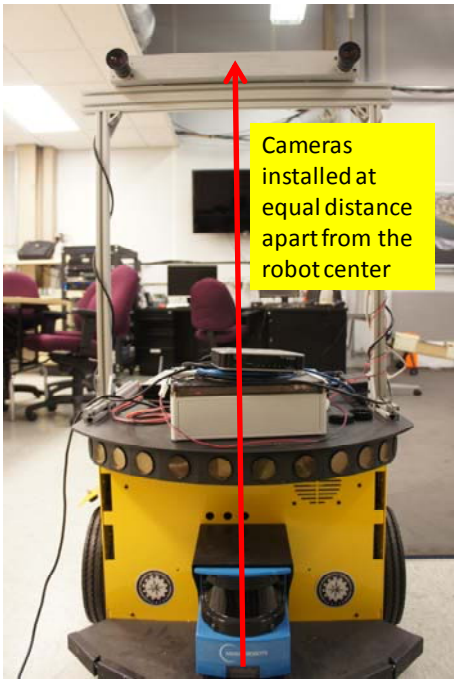

Figure 26: Cameras installed at equal distance from the robot centerline.

3.5.1.5 Frame notations

For clarity and when required, a superscript is added to a coordinate variable to indicate the frame the coordinate variable is represented in: (g) indicates global frame, (c) is camera frame and (r) the robot frame. For example $x^{(g)}$, $x^{(c)}$ and $x^{(r)}$.

3.5.2 Action aiding algorithm

Most of the existing robot's image navigation functionalities and their algorithms (e.g. imaging system, SIFT feature identification, feature matching and landmark localization) are reused as this research focuses on developing a behavioral action aiding engine. This is a good assessment of the feasibility in developing the action aiding engine as a general and light weight algorithm that is applicable to most image navigation solutions without major modifications to their existing core navigation engines. Landmark tracking functionality and behavioral action aiding engine are developed.

3.5.2.1 Landmark tracking

The landmark tracking list maintains awareness of which landmarks can be re-acquired by the robot imaging system even if the landmark is not in the current robot's FOV. This increases the choice of landmarks and potential orientations the robot can take to minimize egomotion errors. At every evaluation instance (i.e. when the robot stops to evaluate the orientation to take next), the landmark tracking list is maintained.

Conversion of a landmark's location in camera frame to global frame

For ease in calculations, landmark locations are represented in the global frame. Equations 25 and 26 describe how a landmark location in camera frame is converted into global frame. First, the angle between the robot's centerline, to the landmark is determined. The distance between the robot and the landmark is also determined.

$$\theta_{lm}^{(c)} = tan^{-1}\left(\frac{dx^{(C)}}{dy^{(C)}}\right) \quad \textbf{(25)}$$

$$dist = \sqrt{dx^{(C)^2} + dy^{(C)^2}} \quad \textbf{(26)}$$

where:

$\theta_{lm}^{(c)}$ = angle between the robot centerline and the line to the landmark.
$dx^{(C)}$ = horizontal displacement from the robot to the landmark.
$dy^{(C)}$ = depth from the robot to the landmark.
$dist$ = absolute distance between the robot and the landmark.

The angle of the landmark from the robot location in global frame is then calculated:

$$\theta_{lm}^{(g)} = \theta_{Ro}^{(g)} - \theta_{lm}^{(c)} \quad \textbf{(27)}$$

where:

$\theta_{lm}^{(g)}$ = angle of the landmark from the robot position in global frame.
$\theta_{Ro}^{(g)}$ = orientation of the robot in global frame.

And finally, the location of the landmark, in global frame is determined:

$$lm\_loc\_x^{(g)} = dist * \cos\left(\theta_{lm}^{(g)}\right) + x_{ro}^{(g)} \quad \textbf{(28)}$$

$$lm\_loc\_y^{(g)} = dist * \sin\left(\theta_{lm}^{(g)}\right) + y_{ro}^{(g)} \quad \textbf{(29)}$$

where:

$lm\_loc\_x^{(g)}$ *and* $lm\_loc\_y^{(g)}$ = coordinates of the landmark location in global frame.
$x_{ro}^{(g)}$ and $y_{ro}^{(g)}$ = x, y coordinates of the robot.

Register new features not in the existing tracking list

Landmarks that are in the robot's FOV, but not in the landmark tracking list are to be registered. A similarity check between the observed landmarks and all the existing

tracking list landmarks is performed via feature descriptor matching, checking the Euclidean distances between the features descriptors. The left camera image feature descriptors of all landmarks in the tracking list are first pre-transposed.

$$desT_{TL} = des_{TL}' \tag{30}$$

where:

$desT_{TL}$ = Transposed feature descriptors of each tracking list landmark.

A current view landmark left image feature descriptor is then dot product-ed with all transposed descriptors from the tracking list and sorted from the smallest to the largest value.

$$dotprods = des_{CV\_lm} * desT_{TL} \tag{31}$$

$$[vals, indx] = sort(\text{acos}\,(dotprods)) \tag{32}$$

where:

$dotprods$ = dot products between the current view landmark left image feature descriptor and all tracking list landmarks' transposed feature descriptors.
des_{CV_lm} = Current view landmark left image feature descriptor.
$desT_{TL}$ = Transposed feature descriptors of each tracking list landmark.
$vals$ = sorted dot product values, from smallest to the largest.
$indx$ = position number of each dot product value before sorting.

The smallest value indicates the descriptor that is the most closely matched (between the current view landmark and a tracking list landmark). If this value is greater than a pre-determined threshold, the current view landmark looks significantly different from all the landmarks in the tracking list and should be registered as a new landmark in the tracking list.

$$\begin{aligned} &if\ (vals(1) > threshold) \\ &\{ \\ &\quad Register\ as\ new\ landmark \\ &\} \end{aligned} \tag{33}$$

Update location, viewpoint angle and descriptor of present FOV landmarks that are in the tracking list

If the smallest Euclidean distance (dotprods value) is smaller than the threshold, there is a possibility that this tracking list landmark is the same as a current view landmark. A series of tests need to be performed to minimize false matching. The first test uses a distance ratio parameter to check that the Euclidean distance to the second feature on the sorted list is clearly farther than to the first. This checks that only one landmark on the tracking list looks uniquely similar to the current view landmark being matched. If the Euclidean distance to the second feature (or even subsequent) is similar (close) to the first value, these two (or more) landmarks in the tracking list look similar to the current view landmark. Ignore the current view landmark and do not declare a match with any landmark in the tracking list. Erroneous matching corrupts features descriptors values and viewpoint angles.

$$\begin{aligned} &if\ \left(vals(1) < distRatio * vals(2)\right) \\ &\{ \\ &\quad Potential\ match \\ &\} \end{aligned} \tag{34}$$

As a second check, a landmark is considered a correct match (between the current view landmark and a tracking list landmark) if their physical locations are within a predetermined threshold area. When a match is declared, the landmark location, viewpoint angle and its left image feature descriptor information in the tracking list are updated. The location determined in the most current image frame is combined with the location stored in the tracking list using weighted averaging. However, for simplification, the location is updated with the data from the current frame only.

$$\delta_{lm}^{(g)} = \sqrt{({x_{lmTL}}^{(g)} - {x_{lmFOV}}^{(g)})^2 + ({y_{lmTL}}^{(g)} - {y_{lmFOV}}^{(g)})^2} \quad \textbf{(35)}$$

$$\begin{aligned} &if\left(\delta_{lm}^{(g)} < threshold\right) \\ &\{ \\ &\quad Update\ location, viewpoint\ angle\ and\ descriptors\ into \\ &\quad the\ tracking\ list \\ &\} \end{aligned} \quad \textbf{(36)}$$

where:

$\delta_{lm}^{(g)}$ = distance between a tracking list landmark location and the current view landmark location in global frame.
${x_{lmTL}}^{(g)}$ and ${y_{lmTL}}^{(g)} = x, y$ coordinates of a tracking list landmark.
${x_{lmFOV}}^{(g)}$ and ${y_{lmFOV}}^{(g)} = x, y$ coordinate of a current view landmark.

Update viewpoint angle of tracking list landmarks that do not appear in the current FOV

To determine if landmarks stored in the tracking list are recognizable after the robot moved, viewpoint angles are updated after each robot step.

$$\theta_{lm}^{(g)} = tan^{-1}\left(\frac{{x_{lmTL}}^{(g)} - x_{ro}^{(g)}}{{y_{lmTL}}^{(g)} - y_{ro}^{(g)}}\right) \quad \textbf{(37)}$$

where:

$\theta_{lm}^{(g)}$= angle of the landmark from the robot in global frame.
${x_{lmTL}}^{(g)}$ and ${y_{lmTL}}^{(g)} = x, y$ coordinates of a tracking list landmark.
$x_{ro}^{(g)}$ and $y_{ro}^{(g)} = x, y$ coordinates of the robot.

If the new viewpoint angle change from when the landmark was last observed is greater than 20°, the landmark is dropped from the list as it may be unrecognizable.

3.5.2.2 Evaluative action aiding decision engine

With the landmark tracking list developed, the key action aiding decision engine algorithm is introduced. The following processes are repeated for each robot step.

Check for landmarks and obstacles in the robot FOV at each angle assessed

Before a particular angle becomes a candidate for the robot to orientate to in the next step, it is necessary to check that landmarks are visible and there are no significant obstacles (e.g. wall) in that orientation. Since the robot is not allowed to rotate physically, the landmark tracking list is used for this assessment. First, the FOV of the robot at the angle under assessment is determined:

$$LFOV^{(g)} = scan\_angle^{(g)} + FOV/2 \quad \textbf{(38)}$$

$$RFOV^{(g)} = scan\_angle^{(g)} - FOV/2 \quad \textbf{(39)}$$

where:

$LFOV^{(g)}$ *and* $RFOV^{(g)}$ = Left and right field of view edges angle of the robot at the particular angle under assessment.

Next the position of the robot is projected virtually ahead by a fixed step size (e.g. 0.25m) in the orientation under assessment (Equations 40 to 43). Equation 37 is used to determine the angles to each landmark from the robot "new" position. Landmarks within $LFOV^{(g)}$ and $RFOV^{(g)}$ angles can be observed. If a landmark is observable after the robot "translates" forward by the step size, it means that a landmark is track-able through the translation and egomotion does not fail. If there are no landmarks in the FOV after the robot is projected ahead, it could mean that there are no track-able landmarks in this orientation and egomotion fails. If there are landmarks in very close proximity of the robot (e.g. slant range < 0.5m), it could mean that the robot is approaching an obstacle and can collide with it. Therefore, this angle is not considered for the next robot step.

This check provides some collision avoidance functionality (for major obstacles like a wall), but should not be used as the only collision avoidance function.

$$\delta_x^{(g)} = step\_size * \cos\left(scan\_angle^{(g)}\right) \quad \textbf{(40)}$$

$$\delta_y^{(g)} = step\_size * \sin\left(scan\_angle^{(g)}\right) \quad \textbf{(41)}$$

$$x_{ro\_new}^{(g)} = x_{ro}^{(g)} + \delta_x^{(g)} \quad \textbf{(42)}$$

$$y_{ro\_new}^{(g)} = y_{ro}^{(g)} + \delta_y^{(g)} \quad \textbf{(43)}$$

where:

$\delta_x^{(g)} and\ \delta_y^{(g)} = x, y$ distance (global frame) the robot changes if it is projected by the step size along the scan angle.
$x_{ro\_new}^{(g)}$ and $y_{ro\_new}^{(g)} = x, y$ coordinate (global frame) of the robot at the new, projected, virtual position.

Obtain egomotion CV for each angle assessed

For each visible landmark within the robot's FOV at the angle assessed, obtain its associated egomotion CV from the error model (look-up table). The displacement ($dx^{(scan)}$ and $dy^{(scan)}$) of the landmark from the robot at the angle under assessment is found through Equations 44 to 48. Using $dx^{(scan)}$ and $dy^{(scan)}$, the egomotion CV associated with a landmark can be obtained from the look-up table. Repeat the same process for all visible landmarks in the FOV at the angle assessed. Thereafter, the combined step egomotion CV for the angle under assessment is obtained using weighted averaging. The above process is repeated for each angle assessed in the robot step. The angle that has the least combined egomotion CV is chosen as the orientation for the robot's next step.

$$\theta_{lm}^{(g)} = tan^{-1}\left(\frac{y_{lmTL}{}^{(g)} - y_{ro}^{(g)}}{x_{lmTL}{}^{(g)} - x_{ro}^{(g)}}\right) \quad \textbf{(44)}$$

$$dist = \sqrt{\left(x_{lmTL}{}^{(g)} - x_{ro}^{(g)}\right)^2 + \left(y_{lmTL}{}^{(g)} - y_{ro}^{(g)}\right)^2} \quad \textbf{(45)}$$

$$\theta_{lm\_Scan}^{(c)} = scan\_angle^{(g)} - \theta_{lm}^{(g)} \quad \textbf{(46)}$$

$$dx^{(scan)} = dist * \sin\left(\theta_{lm\_Scan}^{(c)}\right) \quad \textbf{(47)}$$

$$dy^{(scan)} = dist * \cos\left(\theta_{lm\_Scan}^{(c)}\right) \quad \textbf{(48)}$$

where:

$\theta_{lm}^{(g)}$ = angle of the landmark, from the robot position, in global frame.
$dist$ = distance between robot to landmark.
$\theta_{lm\_Scan}^{(c)}$ = angle of the landmark from the scan orientation.
$dx^{(scan)}$ and $dx^{(scan)}$ = location of the landmark in the scan frame.

3.5.2.3 Action aiding decision engine output

The output is a vector in the camera frame that indicates the rotation angle the robot needs to turn, and the size of the next translation step. If required, a DCM is used to convert the output vector from camera to robot frame. As the camera and robot frames are "aligned" in this research, DCM is not required. This vector is sent to the robot's Unified Behavioral Framework that decides which robot movement behavior (e.g. collision avoidance, Go To, stop etc) is used that best meet the robot overall goal at that moment.

3.5.3 Unified Behavioral Framework

The robot is implemented with a Unified Behavioral Framework (UBF) [28] which modularizes the different possible robot tasks into individually independent behaviors. Each of these behaviors is capable of concurrently generating a set of recommended actions to the higher layer arbiter function, together with a vote field which indicates its desire for selection. Cognizant of the global goals of the system, the deliberator / controller chooses the behavior actions that best serve the system's global goals.

The action aiding engine is developed as a behavior module within the UBF. At each evaluative step, the action aiding behavior module analyzes the latest image and landmarks. A recommended action vector that indicates the angle and step size the robot should make is generated. When the robot moves, images are also continuously collected. During evaluation, the combined robot step (the step just completed) egomotion CV is calculated from all the stored images. The action aiding behavioral module voting field magnitude increases and decreases in direct relation to the egomotion CV.

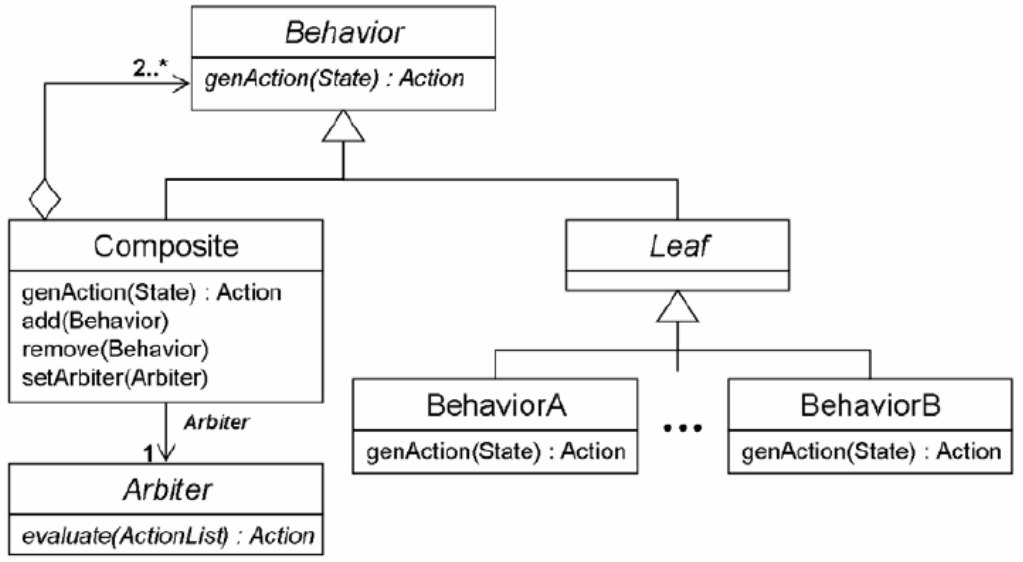

Figure 27: Class diagram for the UBF. (Figure from [28]).

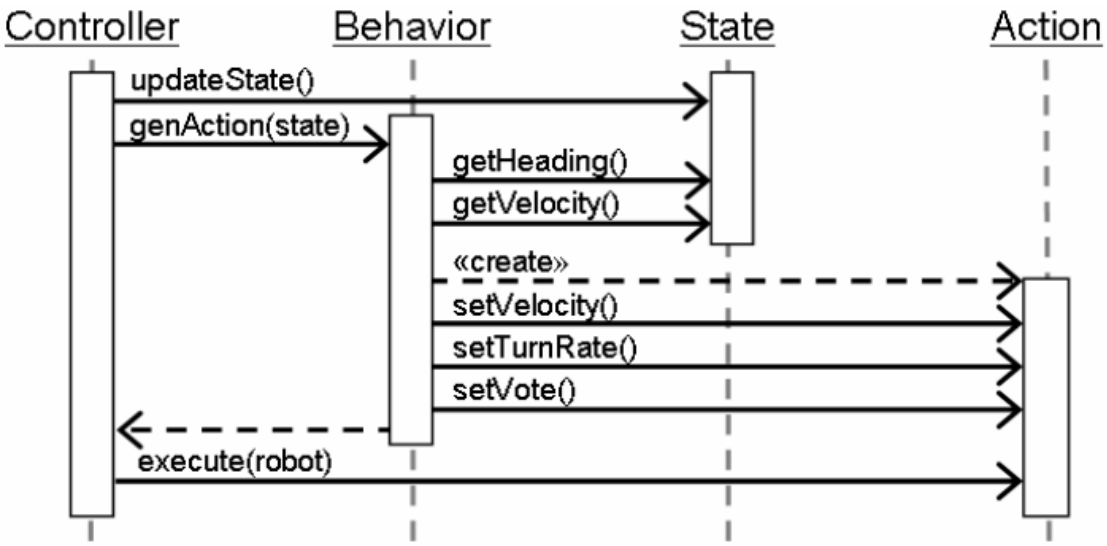

Figure 28: Sequence diagram of a controller using a behavior. (Figure from [28]).

3.6 Action aiding process summary block diagram

Figure 29 summarizes the action aiding processes (see Appendix A for detailed diagram). It also shows how the action aiding algorithm is integrated with the existing robot's UBF.

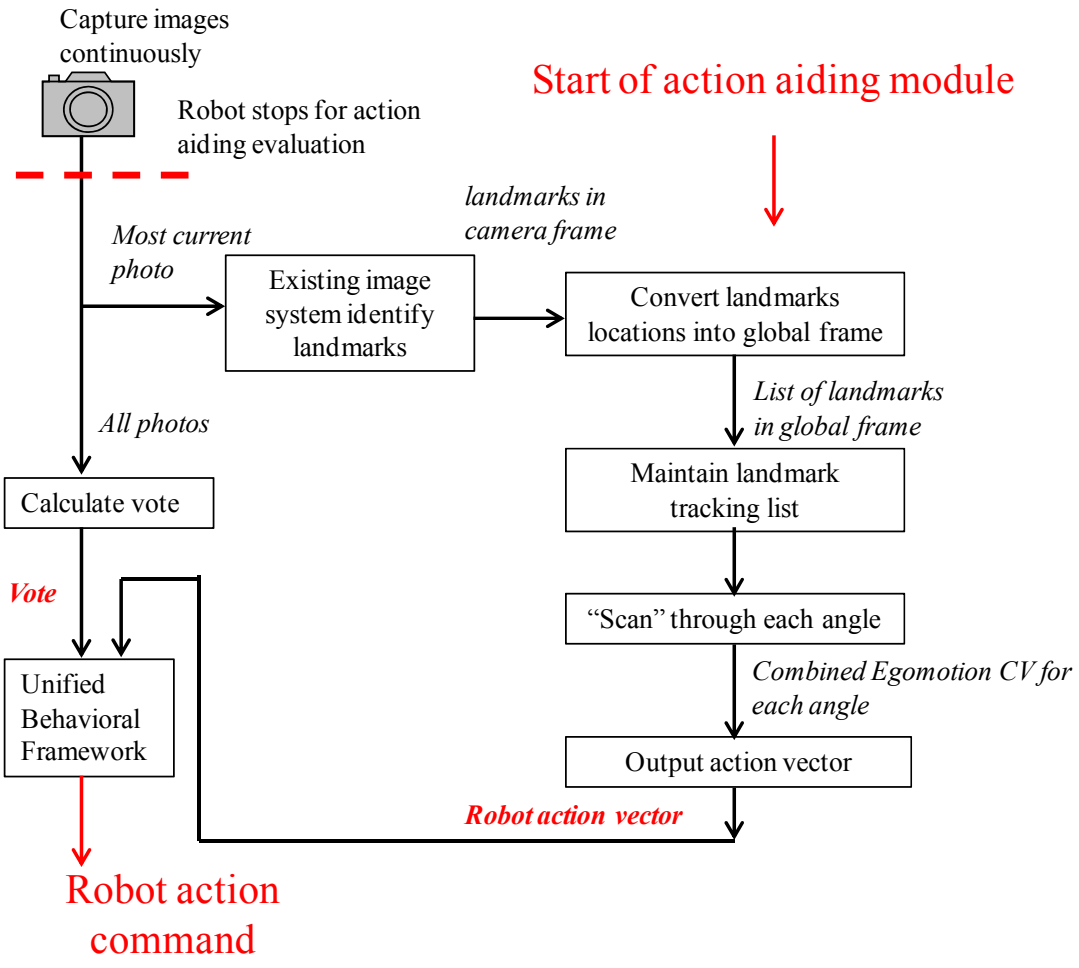

Figure 29: Block diagram depicting the action aiding processes and how it integrates with the robot's UBF.

IV. Analysis and Results

Chapter 3 showed that landmark measurements variation exists which causes egomotion errors. Establishing that egomotion errors are closely related to landmark positions, the landmark location egomotion error model was derived. Observing the distribution of the egomotion CV for different landmark locations in the error model, action aiding techniques were proposed and the algorithm for evaluative action aiding with landmark tracking was developed.

This chapter shows the tests carried out on the various action aiding techniques proposed in Chapter 3. Before performing physical tests using a robot in a real life environment, it is beneficial to understand and compare the operations and performances of the various proposed action aiding techniques in a controlled simulation environment. Evaluation of the results is also presented. Thereafter, the various action aiding techniques are tested on a robot in a controlled indoor environment with various route profiles. The results are tabulated and the performances compared. This chapter also describes the testing equipment, environment and test profiles.

4.1 Test objective

The goal of the research is to develop action aiding techniques that reduce egomotion errors and increase the usability of egomotion in areas with limited features. The test environment and routes are chosen to determine if the action aiding techniques meet these goals. All tests performed in this research compares the relative performances between various action techniques, with “no aiding” profile as the test reference/ benchmark. Emphasis is not placed on the absolute performance of each action aiding technique. An action aiding technique is successful if it gives lesser average egomotion error distance compared to an unaided run using the same route. Likewise, a particular action aiding technique is considered superior to another action aiding technique if it gives lesser average egomotion error distance. In view of this, this research makes

numerous simplifications which does not affect the relative performances, but may affect the absolute egomotion results.

4.2 Proof of concept of the various action aiding techniques

The various action aiding techniques are developed for simulation in MATLAB. Besides being able to analyze the behaviors and relative performances, simulation provides the ability to quickly see the effects of changes made to the algorithm. Having full control within a simulation environment also makes it easier to quantify and compare the various techniques.

4.2.1 Simulation environment

Using MATLAB, a fictitious indoor corridor environment is created. The simulation environment measures 6m x 60m and 16 landmarks are "planted" (shown as red dots) at locations replicating typical landmark positions observed by a robot moving from one end of the corridor. Some landmarks are located along the side walls, while most landmarks are identified at the end of the long corridor. To increase results generality, 10 corridors with randomly rearranged landmark positions are generated (Figure 30). Each action aiding technique (including the baseline reference profile; no aiding) is tested once in each simulated corridor environment. The robot is simulated to travel 40m down the corridors via various action aiding techniques.

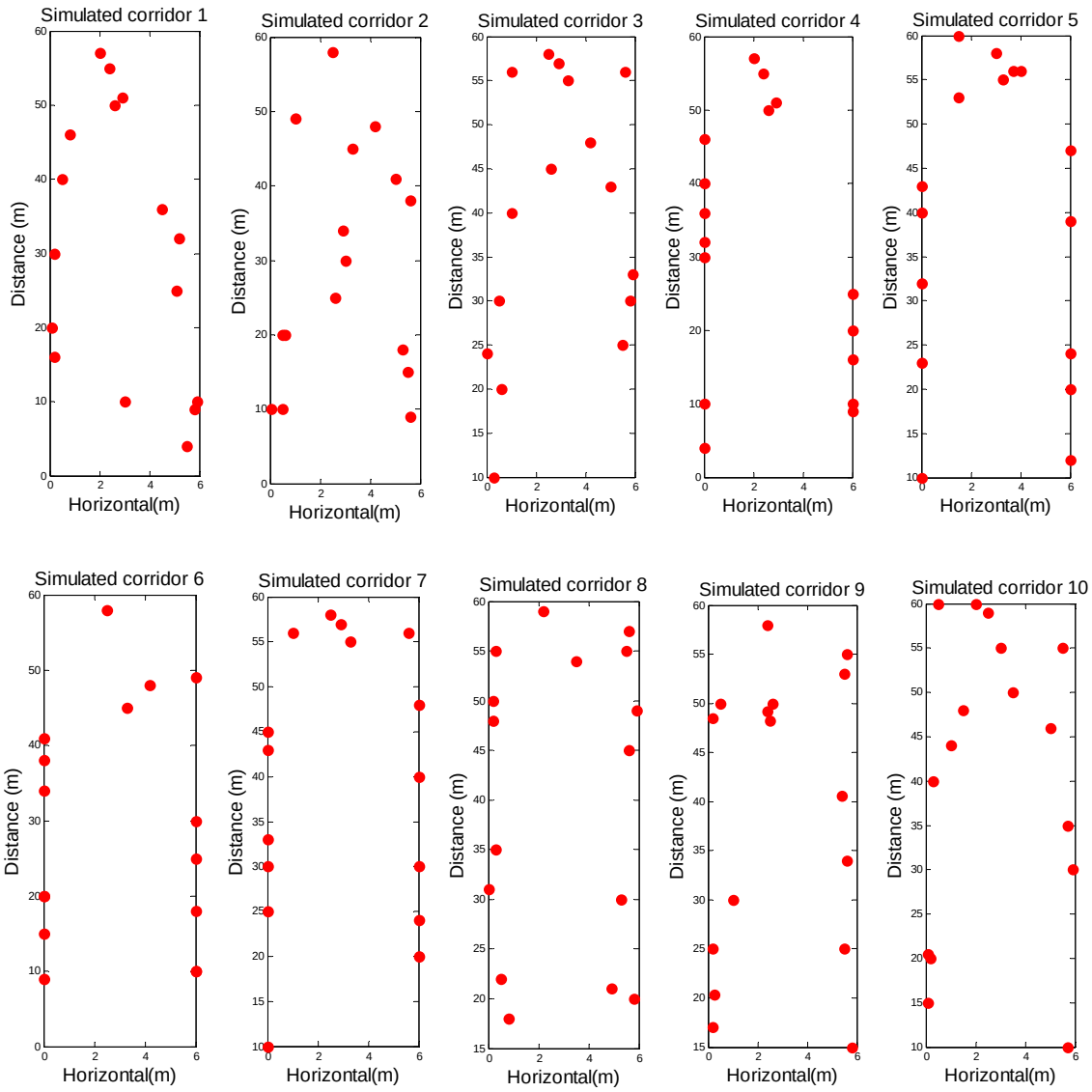

Figure 30: Ten MATLAB simulated corridor environments (6m width x 60m length) with 16 features randomly positioned.

4.2.2 Artificiality and limitation

The simulation algorithm does not take into account that new landmarks can be discovered when the robot observes the same area from different angles. The number of landmarks is much fewer than in a real-world environment. Equipment imperfections (e.g. camera mountings, movement errors, odometry errors etc) are not modeled. Also, the uncertainties and errors associated with identification and matching of features by

SIFT is not modeled. Instead, it depends on the landmark location egomotion error model derived earlier to simulate the relative error contribution from different landmarks. Despite these artificialities, it is a realistic performance comparison tool between the various action aiding techniques, although absolute values are not comparable.

4.2.3 Quantifying simulated action aiding performance

Under simulation conditions, it is not possible to obtain or simulate egomotion error distances since no images are used. Therefore, journey egomotion CV, which is representative of the journey egomotion error, is used for relative performance evaluation of the various action aiding techniques. At each step, the step egomotion CV from landmarks in the robot FOV is determined via the error model (for each landmark) and combined using weighted averaging. Each step's egomotion CV is then combined to form the journey egomotion CV using sum of random variables concept. This combined journey egomotion CV is used to compare the relative performance of various action aiding schemes. Note that the absolute value of this journey egomotion CV is not representative of the true egomotion performance, except for use as a comparison parameter. The smaller this number is, the better the relative performance.

4.2.4 Baseline profile - no action aiding

To determine if the various action aiding schemes reduce egomotion errors, the egomotion errors from each action aiding technique are compared to the error that resulted from a path without action aiding (through the same corridor). Without action aiding, the robot moves in a straight path between the start and end of the corridor. No considerations are given to the placement of the landmarks and the landmarks are used as they appear for egomotion calculation. In MATLAB, the robot is simulated to travel in a straight path. Egomotion CV is calculated for each step and combined for the journey. Figure 31a shows the MATLAB image for one such run. The blue crosses indicate the

positions where the robot makes a decision about the next orientation it should take while the connecting blue lines indicate the path taken by the robot.

4.2.5 Non evaluative action aiding

Instead of travelling straight, the robot is simulated to travel in a “zig zag” manner at angles of 20° from the path center line, regardless of the positions of the landmarks. Figure 31b shows the MATLAB image of one such run.

4.2.6 Evaluative action aiding (physical scans)

In real time, this algorithm seeks out the most favorable orientation for each step (smallest step egomotion CV) taking into consideration the landmark positions relative to the robot. It physically scans every permissible angle before making its decision. Figure 31c shows a typical path taken by a simulated robot implemented with such an algorithm. Notice that the path also has a general "zig zag" shape.

4.2.7 Evaluative action aiding (landmark tracking)

It was highlighted that action aiding with physical scans before each step introduces unacceptable process time. Hence, a landmark tracking list was incorporated to substitute the need for scans. Figure 31d illustrates a path taken when action aided with landmark tracking.

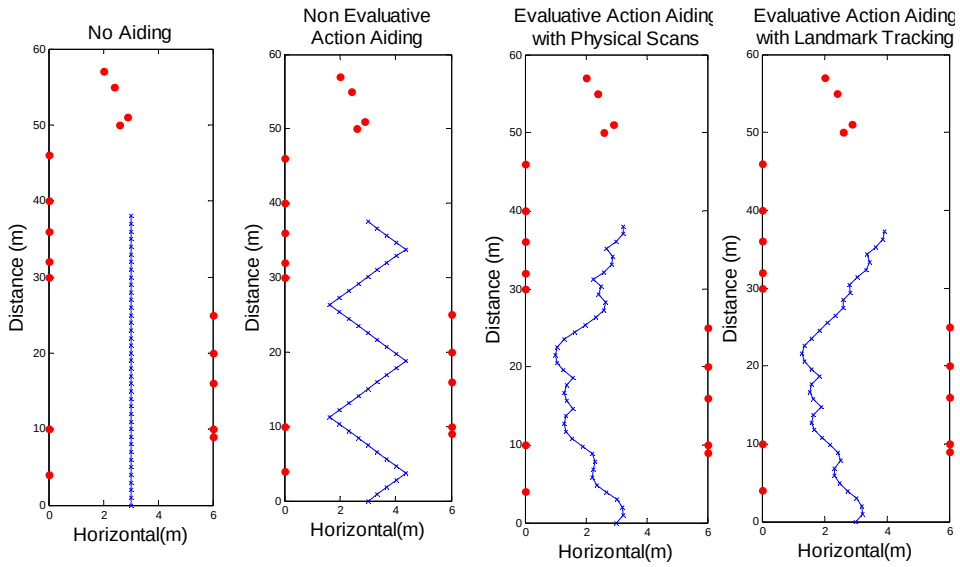

Figure 31: Typical robot path under various forms of simulated action aiding. From left (a) no aiding, (b) non evaluative action aiding, (c) evaluative action aiding (physical scan), and (d) evaluative action aiding (landmark tracking).

4.2.8 Initial Observations

Figure 31 shows the paths taken by the robot through the same simulated environment for different action aiding techniques. Of interest, note the similarity in the paths taken by evaluative action aiding (physical scan) and evaluative action aiding (landmark tracking) techniques. The tracking list was developed to allow the robot to maintain awareness of landmarks not in the current robot FOV, without physically rotating. Similarity of the two paths illustrates that the tracking list is successful in replicating the physical scans performed by evaluative action aiding (physical scan). The journey CV was also very close, with evaluative action aiding (physical scan) scoring 295 and evaluative action aiding (landmark tracking) scoring 321 (lower value is better).

4.3 Simulation results

Table 4 shows the journey egomotion CV for each action aiding technique. Note that the journey egomotion CVs are for relative performance comparison. With a smaller journey CV, lesser egomotion errors are expected.

Table 4: Simulation results – Journey egomotion CV with 10 simulation runs for different action aiding techniques.

Run #	No aiding	Non evaluative aiding	Evaluative action aiding with physical scans	Evaluative action aiding with landmark tracking
1	859	670	291	335
2	1560	606	381	491
3	1267	534	416	421
4	721	738	295	321
5	1012	626	396	489
6	668	541	291	329
7	1012	590	430	500
8	1572	661	541	612
9	1282	587	503	581
10	1242	705	404	485
Average journey CV	**1120**	**626**	**395**	**456**
Improvement (%)	**Baseline**	**44%**	**65%**	**59%**

With action aiding, journey egomotion CV is smaller compared to no aiding, hence egomotion errors are expected to be reduced. Of the 3 action aiding techniques, non evaluative technique performs the worst (44% improvement) as it lacks the ability to observe landmark positions. Evaluative action aiding (physical scan) which physically scans the environment has the most complete knowledge of landmark locations and hence, it is the best performing (65% improvement) action aiding technique. Evaluative action aiding (landmark tracking) performs between non evaluative action aiding and evaluative action aiding (physical scan) techniques (59% improvement). It is also observed that non evaluative action aiding (landmark tracking) faired only slightly worse

compared to evaluative action aiding (physical scan) and yet provides a major advantage in the journey / process speed.

4.4 Physical test

Simulation results show that action aiding has the potential to reduce egomotion uncertainty and errors. The various action aiding techniques are next physically tested using a robot in a controlled environment. This section provides information of the test equipment, environment and the profiles. The conduct of the test is described and evaluation approach defined. As post analysis software is required to determine the egomotion path, the algorithm development is also presented. Before the conduct of the actual tests, evaluation runs were performed and the relevant issues discovered during these runs are highlighted. Thereafter in the next section, the test run results are presented and detailed analysis shown.

4.4.1 Test equipment - Robot

The robot used in the test is the Mobile Robots, Inc. Powerbot equipped with stereo camera system, odometry, SICK laser scanning unit (Lidar), Inertial Measurement Unit (IMU) and ultrasonic distance measurement system (Figure 32). The Lidar, IMU and ultrasonic distance measurement system are not used in this research. When manual control of the robot is required, the externally attached controller is used. The test algorithm is implemented within the UBF [28], which together with all other software components, are installed in a laptop which physically integrates all systems and sensors.

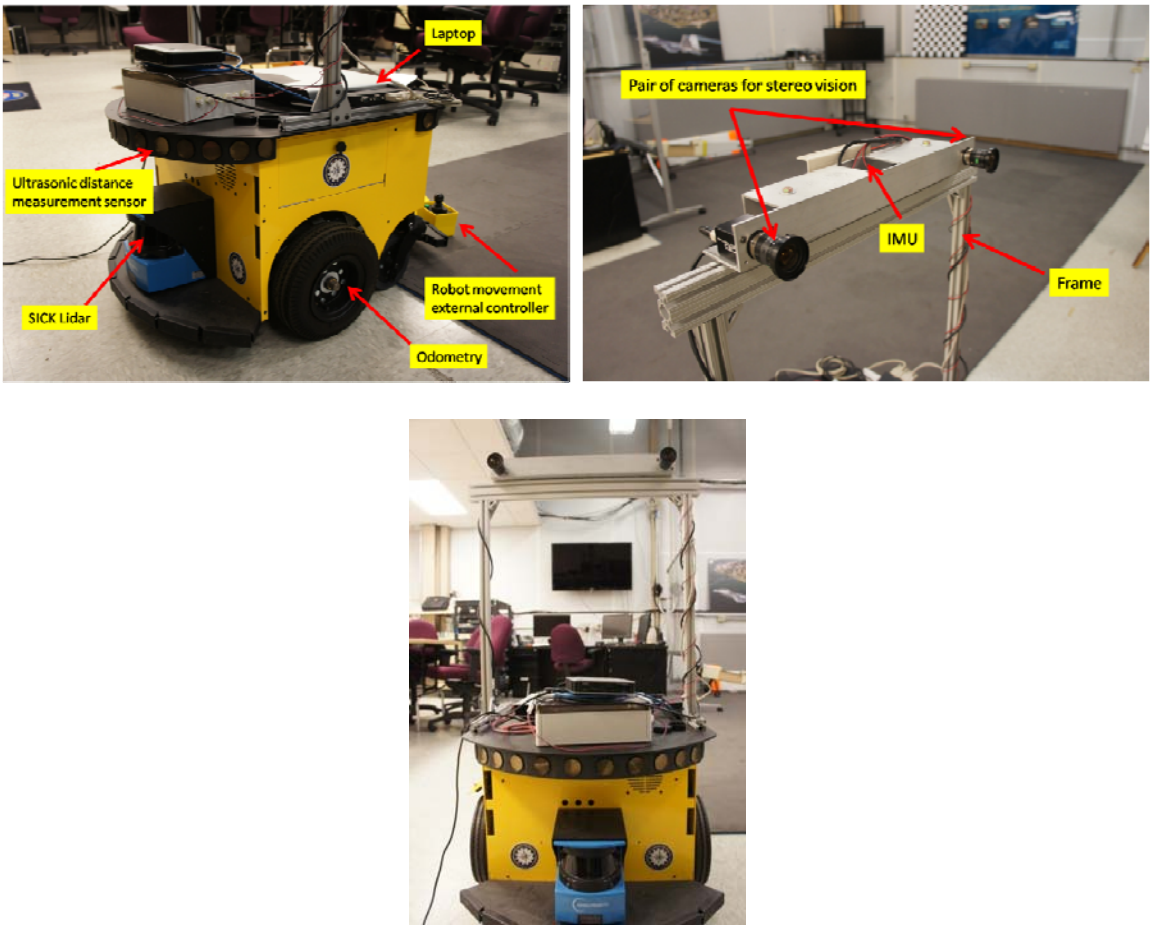

Figure 32: Pictures showing the various components of the robot.

During trials, it was discovered that when the robot was commanded to move straight, it skewed slightly to the right. As there was no opportunity to calibrate it, only the tire pressure was adjusted for compensation. When driven in manual mode, it is corrected by making small direction adjustments. Although it did not skew significantly, it affects the effectiveness of evaluative action aiding in autonomous modes as the vehicle does not translate or rotate to the positions / angles determined by the action aiding algorithm. This error is to be taken into consideration when comparing egomotion errors from different action aiding techniques.

4.4.2 Test objectives, environment and routes

To recap, this research aims to develop action aiding techniques that reduce egomotion errors (compared to no aiding) and increase the usability of egomotion in areas with few image tracking points (i.e. limited features). Therefore, the test environment and routes are specifically chosen to determine if the various action aiding techniques meet these goals.

Referring to Figure 33 and 34, AFIT building 640 level 2 corridor was chosen as the test route. The straight path between locations 1 and 2 is used (both directions are used). The distance between locations 1 and 2 is 40.733m. A section of this route, from the midpoint of locations 1 and 2 to location 1 (20.37m) that has a sizable featureless wall was chosen for limited feature area testing (a recycling bin was shifted to create a bigger area with few features). Only the direction towards location 1 is used as the other direction is long and many features exists.

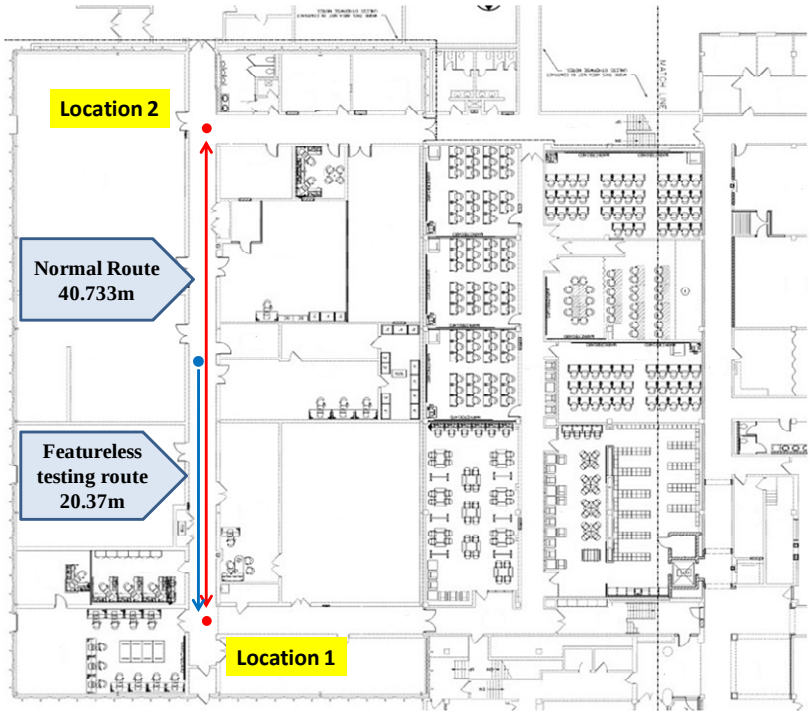

Figure 33: Test area and routes.

Figure 34: Sample image of the test routes. Top: Location 1 to 2. Middle: Location 2 to 1. Bottom: Midpoint of location 1 and 2 to location 1.

From earlier analysis of the effects of landmark distances on egomotion errors, it is expected that egomotion errors in the direction of location 1 to 2 are greater than from location 2 to 1. This is because location 1 reaches the end of the corridor and landmarks on the exit door are nearer. However beyond location 2, there is a distance through another lobby before it reaches another set of doors. Hence landmarks are farther (i.e. the corridor looks longer despite the same distance moved by the robot).

4.4.3 Quantifying action aiding performance in physical tests

In a test run, the robot navigates from a known start point (e.g. location 1) to a predetermined destination point (e.g. location 2) under the influence of an action aiding technique being tested. Images are post processed to determine the egomotion perceived stop position. The error distance which is defined as slant distance between the true stop position and the egomotion perceived stop position (|true stop position – egomotion perceived stop position|) is determined. The test is repeated and the mean egomotion error distance for each action aiding technique is obtained. The action aiding technique that gives the least mean egomotion error distance is the most successful. Standard deviation of the egomotion error distances for each test profile (test route direction and action aiding type) is also determined. For no aiding (straight paths) and evaluative action aiding profiles, the egomotion error distance standard deviation is expected to be small, while non evaluative action aiding egomotion error distance standard deviation is expected to be large.

Note that only the relative egomotion error distance is evaluated as the absolute egomotion error distance derived in this research is not representative of the true performance due to the many simplifications carried out in the implementation. These simplifications include using the simplified egomotion calculation algorithm, no additional processing is done to remove outliers, or handle other feature matching errors. Since egomotion algorithm is not running real-time on the robot, egomotion is not used to guide the robot. Instead odometry is used for robot translation and rotation measurements. The errors associated with odometry measurements are not corrected.

4.4.4 Tests image collection techniques

No aiding and non evaluative action aiding

The robot collects images while navigating. The images are post processed to determine the egomotion paths. Between the true start and end positions, the robot is manually driven in a straight line (at 0.6m/sec) to simulate unaided navigation. For non evaluative action aiding, the robot is driven manually in a "zig zag" path along the corridor between the start and end positions. The turning positions are not at fixed points to simulate the non evaluative nature of this action aiding technique.

Simulated evaluative action aiding

Implementation issues prevented meaningful testing of autonomous image navigation with evaluative action aiding (landmark tracking). Therefore, a simulated evaluative action aiding was tested instead. At each robot step, multiple image frames for each scan angle are processed offline to determine the average egomotion CV value for that angle. The angle that gives the lowest average egomotion CV is chosen for the next robot step. The robot is rotated to the chosen angle and driven for a fixed step size and the process is repeated again until the robot reaches the true destination point. The process resembled evaluative action aiding (physical scan) but simulation results shows that the performance for evaluative action aiding (landmark tracking) and evaluative action aiding (physical scan) do not differ much. Hence, the results for evaluative action aiding (landmark tracking) collected via this method would provide a fair relative performance comparison against non evaluative action aiding and no aiding. However, since the robot is now manually driven, the poor accuracy and consistency in rotating / moving the robot to the required angles and position would slightly compromise the evaluative action aiding (landmark tracking) performance. The egomotion error distance standard deviation is also expected to be larger than possible.

4.4.5 Simplified robot movements

With concerns that high processing loads for real time image processing and action aiding causes delays in the output of the action aiding vector resulting in accumulating errors, it is decided that sufficient stop time between robot movements be incorporated to allow action aiding computations to complete. The various robot movements are also decoupled to farther reduce egomotion complexity thus reducing computational load (Section 4.4.7). Hence, the robot adopts a move-stop-move and rotate-stop-translate (hence, rotation and translation movements are decoupled) movement profile. Effectively, it moves in steps of a single motion type with evaluative time between steps.

4.4.6 Post processing algorithm

An algorithm was developed to post process collected images to determine the egomotion path from various runs. While this is necessary for test and evaluation purpose (determine the egomotion error distance), it is not required for the actual operation of the action aiding engine in the robot. However, if desired, the algorithm can be implemented in the robot (with minor modifications) to provide near real-time egomotion information. Unfortunately, the existing robot does not have enough processing power.

This section presents a practical implementation of the theory discussed in Section 2.3 on image navigation. From each image, SIFT features are identified (Section 2.3.2) and landmarks determined from each corresponding left and right images of a frame (Section 2.3.3). The locations of each landmark is then determined using epipolar geometry calculation (Section 2.3.4). Between successive frames, corresponding landmarks are identified (using the same “matching” technique covered in Section 2.3.3) so that its relative movements can be used to calculate robot egomotion (Section 2.3.5).

Due to the stochastic nature of measurements and possibility of erroneous feature association between left and right images, and wrong matching of landmarks between successive frames, egomotion value outliers will occur. Advance image processing techniques can be performed to reduce false positive features / landmarks matches. For example, INS data can be used to estimate the robot's movement and limit the search for the corresponding landmark at the next frame within the expected area the same landmark could be located at given the movement of the robot. However, this research concentrates on pure egomotion and thus, all non imaging systems assistance are excluded. Simple methods based on the knowledge of some physical properties of the robot are implemented to reduce errors. Since the physical characteristics of the robot are known, bounds can be implemented to exclude impossible values. Firstly, the robot maximum speed is reduced (0.6m/s) to minimize descriptor changes (for the same feature) between frames, facilitating easier matching. Secondly, features that indicate speeds beyond the maximum known travel speed of the robot (set at 0.6m/sec) are ignored. Thirdly, features that indicate negative speeds are also ignored as the robot in the experiments only travel forward and therefore negative speeds / distance of travel (i.e. travelling in reverse) is not possible. The second and third method also reduces (to a certain amount) the egomotion inaccuracy effects caused when there are non static features (with large speed differential with respect to the robot) within the frames. For example, a person walking (faster) away from the robot shows up as negative speeds. Hence, the effects from these features are ignored. A person walking towards the robot appears as speeds greater than the fastest speed of the robot (if the robot is already moving at the fastest speed). These too are ignored.

4.4.7 Simplified 2D egomotion algorithm

With the simplified (decoupled) robot movement profile and being a ground based vehicle, it is not necessary to use full 3D (6 Degree Of Freedom) egomotion algorithms described in Section 2.3.5. The iterative computational approaches are both processing load and time consuming. Therefore, a simplified 2D (translation, rotation)

algorithm is developed and implemented in place of a full algorithm. Since the research focuses on comparing relative performances between the various action aiding techniques and not absolute egomotion error values, using the simplified egomotion algorithm meets project objectives. It is developed with the following assumptions: (1) landmarks are static and observed landmark position change is only caused by robot movements. This removes the need to account for moving objects. (2) Robot movements are decoupled. It either translates or rotates, but never both together. This allows rotation and translation to be calculated separately. (3) Intervals between actions (e.g. stop-rotate-translate) is longer than 1 image frame period (set at 0.5 secs). This allows an action state to be completely captured by successive image frames. Consider a single observed landmark and its corresponded landmark in a successive frame. Their locations are related by Equation 49.

$$P' = R_w P + T \tag{49}$$

where:

P and P' = the corresponded landmark locations in successive frames.
T = translation.
R_w = rotation angle.

By determining the distances (*dist*) to the same corresponded landmark between successive image frames (Equation 50), it is possible to determine if a robot translated. If $|P'| \approx |P|$, translation (T) is zero since rotation (R_w) changes angles but not magnitude. As the polar angle of P' and P are related by R_w, the robot rotation angle is determined if the difference in polar angle of P' and P is known. If there are no differences in the polar angles, the robot had been stationary. On the other hand, if $|P'| \neq |P|$, then translation occurred and its magnitude (T) is the distance between P' and P [29].

$$dist = abs\left[\sqrt{{dx_i}^{(c)^2} + {dy_i}^{(c)^2}} - \sqrt{{dx_{i+1}}^{(c)^2} + {dy_{i+1}}^{(c)^2}}\right] \tag{50}$$

where:

${dx_i}^{(c)}$ and ${dx_{i+1}}^{(c)}$ = horizontal distance to the landmark at frame i and $i+1$.
${dy_i}^{(c)}$ and ${dy_{i+1}}^{(c)}$ = depth from the robot to the feature at frame i and $i+1$.

Translation

A translation took place if there is a distance change to the same landmark between subsequent frames (a threshold was incorporated to minimize erroneous interpretation due to noise. i.e. $dist > threshold$). Geometrically, the translation magnitude along the current heading is the change in depth:

$$T = abs\left(dy_i{}^{(c)} - dy_{i+1}{}^{(c)}\right) \quad \textbf{(51)}$$

Rotation

If $dist < threshold$ (i.e. no change in the distance), the robot is either stationary or rotating. The angle rotated is determined through Equations 52 to 54. If there is no change in angle, the robot is stationary.

$$\theta_{lm\_i}{}^{(c)} = tan^{-1}\left(\frac{dx_i{}^{(c)}}{dy_i{}^{(c)}}\right) \quad \textbf{(52)}$$

$$\theta_{lm\_i+1}{}^{(c)} = tan^{-1}\left(\frac{dx_{i+1}{}^{(c)}}{dy_{i+1}{}^{(c)}}\right) \quad \textbf{(53)}$$

$$\theta_{rotated} = \theta_{lm\_i}{}^{(c)} - \theta_{lm\_i+1}{}^{(c)} \quad \textbf{(54)}$$

where:

$\theta_{lm_i}{}^{(c)}$and $\theta_{lm\_i+1}{}^{(c)}$ = angles between the robot's center line to the landmark (positive angle represents to the right, and negative angle to the left of the line) at image frame i and $i + 1$ respectively.

Determine unified robot action

The above calculations determine the egomotion value $(T, \theta_{rotated})$ from one landmark. Within a single frame, there can be multiple landmarks. Due to measurement uncertainties, each landmark gives a slightly different egomotion value. Besides

movement magnitude, landmarks can also interpret different movements; stationary, rotation or translation. It is necessary to have a single decision for both movement and its magnitude for each frame, from all available landmarks within the same image frame. Weighted averaging is used to decide the movement type and its magnitude. Landmarks from each frame are consolidated into the various movements they represent. From the landmark position egomotion error model, the CV associated with each landmark is obtained. Using weighted averaging, the combined egomotion CV for each movement category is calculated. The category with the smallest egomotion CV is chosen as the movement type for the frame. Again using weighted averaging, the best estimate of the movement magnitude is computed from the landmarks within the chosen movement category. The egomotion movement type and its magnitude for each frame are thus determined ($T_{frame}, \theta_{rotated\_frame}$).

Robot pose update

The translation and rotation magnitudes described above are in camera frame. To determine the robot's position in the navigation space, it is more convenient if it is described in the global frame which also makes it easy to plot the egomotion path.

If translation took place, the change in robot location in x and y coordinates in global frame ($dx^{(g)}\ dy^{(g)}$) between frames i and $i+1$, are:

$$dx^{(g)} = T_{frame} * \cos(\phi^{(g)}) \quad \textbf{(55)}$$

$$dy^{(g)} = T_{frame} * \sin(\phi^{(g)}) \quad \textbf{(56)}$$

where:

$\phi^{(g)}$ = robot's current orientation / heading in global fixed frame.

The updated robot's location ${x_{Ro\_new}}^{(g)}$ and ${y_{Ro\_new}}^{(g)}$ in global frame is:

$$ {x_{Ro\_new}}^{(g)} = {x_{Ro}}^{(g)} + dx^{(g)} \quad \textbf{(57)} $$

$$ {y_{Ro\_new}}^{(g)} = {y_{Ro}}^{(g)} + dy^{(g)} \quad \textbf{(58)} $$

where:

${x_{Ro}}^{(g)}, {y_{Ro}}^{(g)}$ = current robot location in global frame.

If a rotation took place, the new robot orientation (${\theta_{Ro\_new}}^{(g)}$) in global frame is:

$$ {\theta_{Ro\_new}}^{(g)} = {\theta_{Ro}}^{(g)} - \theta_{rotated\_frame} \quad \textbf{(59)} $$

where:

${\theta_{Ro}}^{(g)}$= current robot orientation in global frame.

4.4.8 Implementation tuning

The numerous implementation simplifications in this research generated significant errors in the resultant egomotion absolute values. Although this does not affect the comparison of the relative performances between the various action aiding techniques (which is the objective of this research), it is desired to minimize the absolute errors if feasible. Therefore, implementation correction parameters can be incorporated in the simplified 2D egomotion algorithm. The rotation parameter is derived by physically rotating the robot through known angles. The images are then processed to determine the egomotion rotation angles. The rotation compensation (RC) parameter is then computed using Equation 60. Using a similar approach, the translation compensation (TC) is computed using Equation 61. The compensation parameters are then multiplied to the respective movements determined by the simplified 2D egomotion algorithm. However, these compensation parameters are not tuned and applied. Instead, they were set to 1.

However, this does not affect the results for relative performance comparison. For future work, this tuning can be incorporated.

$$RC = \frac{Actual\ rotation\ angle}{Egomotion\ derived\ rotation\ angle} \quad \textbf{(60)}$$

$$TC = \frac{Actual\ translation\ distance}{Egomotion\ derived\ translation\ distance} \quad \textbf{(61)}$$

4.4.9 Extending simplified egomotion algorithm to full egomotion

The various action aiding techniques address the fundamental landmark locations dependent errors for egomotion calculation and apply to any egomotion algorithm. Therefore, implementing the full egomotion algorithm does not affect the relative performances determined in this research. The results presented here using the simplified 2D egomotion algorithm is representative of the relative performances of the various action aiding techniques if the full egomotion algorithm is used.

4.4.10 Evaluation run issues

Before the conduct of actual tests, the robot was driven along the test route to determine if there were unexpected test environment and profiles issues. Relevant issues are highlighted.

4.4.10.1 Minimum distance for stereo FOV

As the cameras are laterally displaced with their own FOVs, there is a minimum separation distance between the features and the cameras for stereo FOV to be available. For illustration, assume the inter-camera distance is 0.6m and the FOV of each camera is 90°. The minimum theoretical distance for a feature to be seen in both cameras is 0.3m

(Figure 35). In practice, the features have to be even farther away. If the FOV is narrower or the cameras are separated farther, the minimum stereo FOV distance increases. Figure 36 illustrates 2 situations where the robot is close to the wall, limiting the availability of common features in both left and right camera images.

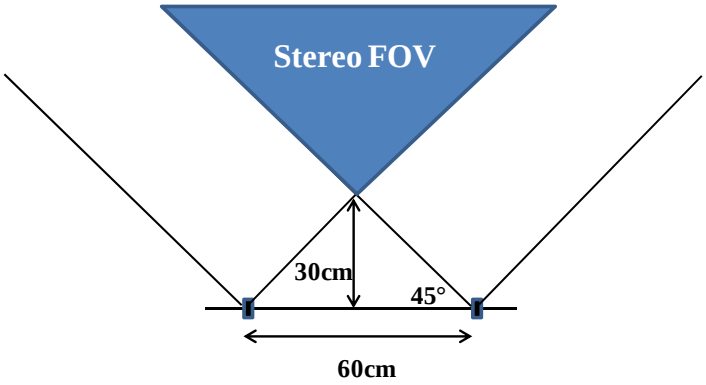

Figure 35: Minimum distance between features and camera for stereo FOV.

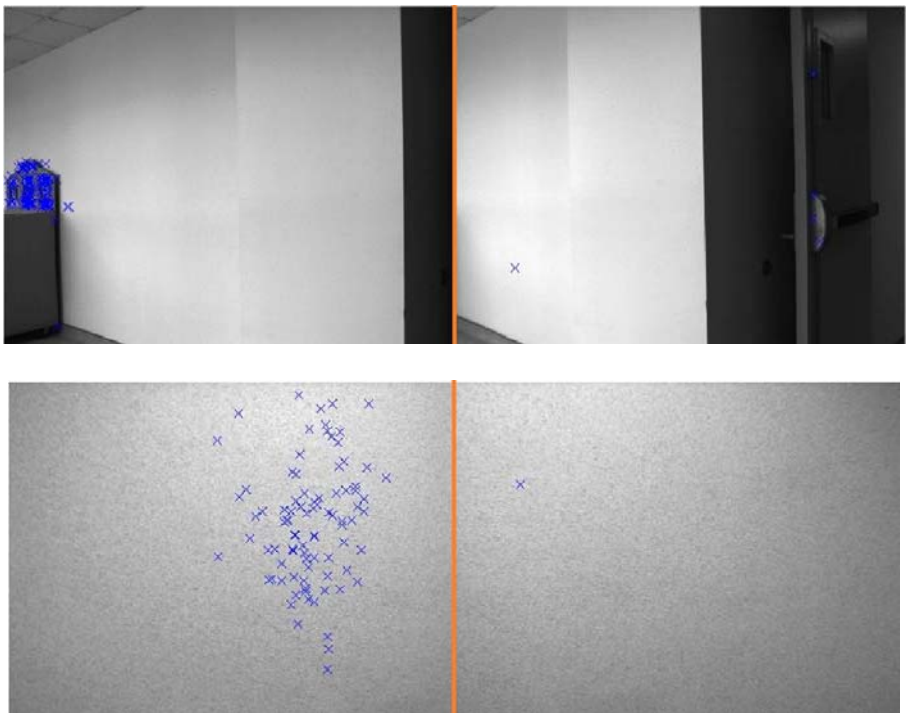

Figure 36: Figures showing 2 situations when the robot is too close to the features. The left and right cameras could not observe the same features.

4.4.10.2 Featureless wall

Besides meeting the minimum distance for stereo FOV, SIFT features must also be available within this limited FOV. Figure 37 shows the stereo FOV dimension (looking at 1 dimension; horizontal) at various distances from the robot.

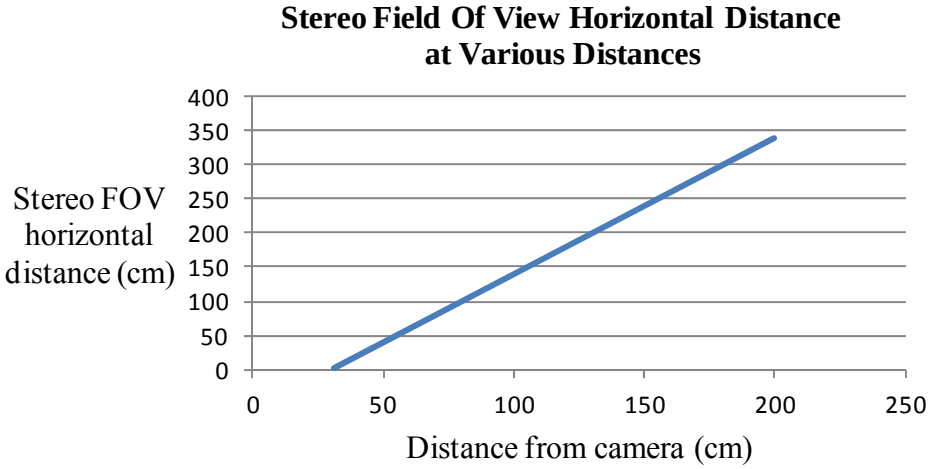

Figure 37: Chart illustrating the stereo FOV dimensions for various distances from the cameras.

Given the dimensions of the corridor (the width is 2.4m), the robot is typically 1m away from the wall. From the above chart, the stereo FOV horizontal dimension is 140cm. Given that walls are relatively featureless, it is difficult to ensure that there are SIFT features within this 140cm at all times, especially while rotating. Figure 38 illustrates a situation when no features are identified on a plain featureless wall at close distance to the robot (blue crosses would have been marked against identified features). The data is captured while the robot is turning at a corridor corner. In such a situation, egomotion fails. This illustrates the difficulty of pure egomotion when turning within tight confines near featureless walls.

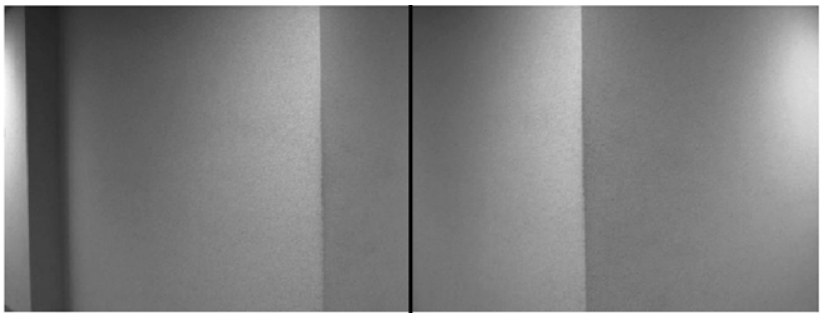

Figure 38: SIFT did not identify any features against a plain wall.

4.4.10.3 View point angles issues

The closer a feature is to the cameras, the larger the viewpoint angle difference when viewed from the left and right cameras. When the viewpoint angle difference is too large, the descriptor for the same feature begins to differ, affecting successful matching of that same feature between the left and right camera. As seen in Figure 3, the probability of correctly matching the descriptor belonging to the same feature falls below 80% when the viewpoint angle difference is beyond 30° [21]. Figure 39 shows that the feature must be at least 110cm away from the cameras to ensure that the viewpoint to the same feature from the left and right cameras remains below 30°.

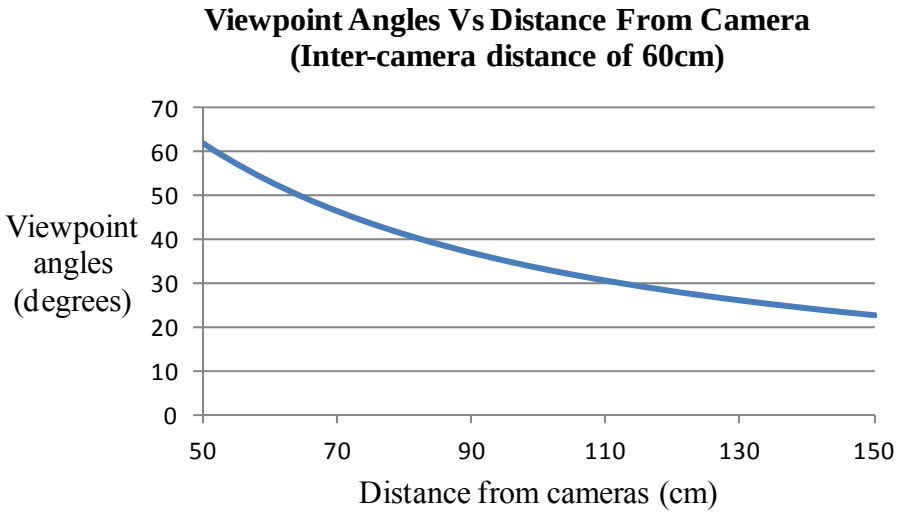

Figure 39: Viewpoint angles for various distances from the camera (Inter-camera distance of 60cm).

Figure 40 illustrates an instance where the features are too close to the cameras and viewpoint angles differed too greatly. Although there are visually similar features identified separately by the left and right cameras, the algorithm determined that no features can be matched between the left and right images as the descriptors differed too greatly (a line would be drawn connecting matched features if there were).

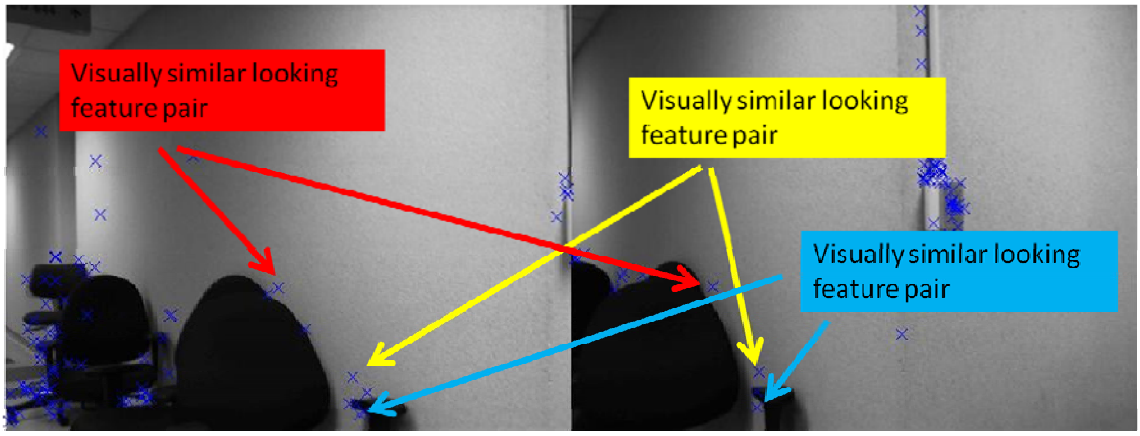

Figure 40: Left and right camera images illustrating viewpoint angle issues.

4.4.10.4 Presence of non-static features

In one of the preparatory runs, a person walked towards the robot. The presence of this non static feature significantly corrupted the robot egomotion data (Figure 41). Notice that the egomotion path in red is much shorter than the true path (from location 2 to location 1). This reinforces the requirement that all landmarks must be static. Although additional processing methods such as RANSAC (Random Sample Consensus) can be incorporated to mitigate the effects, it is not incorporated in this research.

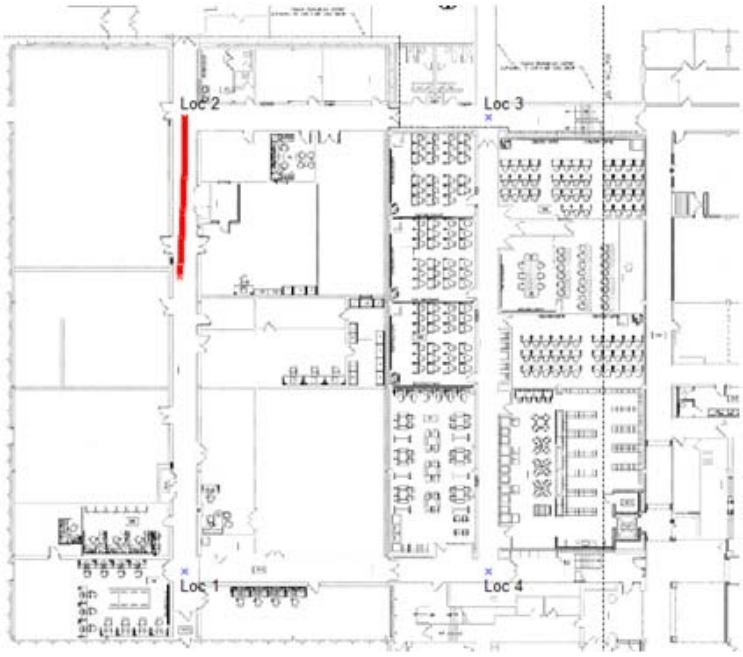

Figure 41: Effects of non static features on egomotion.

4.4.10.5 Relevant lessons for action aiding

Observations from Sections 4.4.10.1 through 4.4.10.4 are relevant lessons for action aiding. In the course of action aiding, the robot should not navigate at large angles from the longitudinal path as it will face the walls too "squarely", increasing the probability of not identifying any features. The robot should not move too close to the walls as it will lose its stereo FOV (hence there is a need to implement a wall proximity sensing and avoidance functionality). In addition, it is necessary to ensure the environment is free of non static features.

4.5 Physical test results and analysis

Tables 5, 6 and 7 list the various test results. The tables show the egomotion error distances in meters (the distance between the egomotion perceived stop position, and the ground truth marker position) for each test run.

Table 5: Test run egomotion error distances – From location 1 to 2 (normal route).

Run #	No aiding (baseline)	Non evaluative action aiding	Simulated evaluative action aiding
1	11.7758m	7.451m	8.1089m
2	13.1734m	10.6805m	0.862m
3	15.2304m	6.5046m	4.4321m
4	15.8832m	12.006m	7.4936m
5	16.4186m	13.5662m	4.4108m
6	14.8256m	4.551m	3.4373m
7	11.4959m	10.8883m	8.3401m
8	14.3034m	9.5733m	0.8223m
9	15.2907m	16.0899m	1.2828m
10	12.2907m	8.9636m	0.2587m
Average egomotion error distance	**14.0424m**	**10.0274m**	**3.9448m**
Std Dev	**1.7989m**	**3.4113m**	**3.1603m**
Error reduction (%)	**N.A.**	**29%**	**71%**

Table 6: Test run egomotion error distances - From location 2 to 1 (normal route).

Run #	No aiding (baseline)	Non evaluative action aiding	Simulated evaluative action aiding
1	8.4988m	4.7381m	0.7925m
2	9.3194m	8.4107m	1.7443m
3	9.129m	5.1936m	1.0588m
4	9.6048m	4.9951m	0.5809m
5	7.1836m	5.1532m	4.8668m
6	7.129m	2.5219m	1.4959m
7	7.8083m	4.7901m	3.4490m
8	7.59m	5.1535m	1.3950m
9	11.669m	8.1235m	2.1134m
10	8.458m	7.9379m	4.8562m
Average egomotion error distance	**8.6389m**	**5.7017m**	**2.2353m**
Std Dev	**1.3766m**	**1.8688m**	**1.5978m**
Error reduction (%)	**N.A.**	**34%**	**74%**

Table 7: Test run egomotion error distances - midpoint to location 1 (limited features area test).

Run #	No aiding	Simulated evaluative action aiding
1	9.805m	2.1099m
2	7.0324m	1.0237m
3	9.3091m	0.5848m
4	9.8124m	2.6417m
5	9.2624m	3.2065m
6	9.4184m	3.5203m
7	9.7299m	2.7758m
8	9.8631m	3.2164m
9	9.0523m	2.0268m
10	8.5701m	1.3442m
Average egomotion error distance	**9.18551 m**	**2.2401 m**
Std Dev	**0.85894 m**	**1.00326 m**
Error reduction (%)	**N.A.**	**76%**

4.5.1 Reduced egomotion error distances with action aiding

Compared to baseline (no action aiding), non evaluative action aiding reduced egomotion error distance by an average of 31.5%, while simulated evaluative action aiding reduced egomotion error distance by an average of 72.5% (average of the normal routes from location 1 to 2 and from location 2 to 1). This result clearly illustrates the effectiveness of action aiding, especially evaluative action aiding (landmark tracking) in reducing egomotion error distances from the unaided straight path. This result demonstrates the successful accomplishment of one of this research's goal: to develop a robot action aiding technique that reduces egomotion errors compared to the unaided robot movement. Table 8 summarizes the egomotion error distances accomplished by the various action aiding techniques compared to no aiding. The improvement percentage, compared to no aiding, is also shown.

Table 8: Average egomotion distance errors and improvements percentage (compared to the egomotion error distance from no aiding) for the various action aiding techniques.

Test route: Location	No aiding (baseline)	Non evaluative action aiding		Simulated evaluative action aiding	
1 to 2	14.04m	10.02m	29% improvement	3.94m	71% improvement
2 to 1	8.63m	5.70m	34% improvement	2.23m	74% improvement

4.5.2 Different egomotion error distance standard deviation for different action aiding types

Referring to the normal route tests (location 1 to 2 and location 2 to 1), the egomotion error distance 1-σ standard deviation for each action type (no aiding, non evaluative action aiding and simulated evaluative action aiding with landmark tracking) shows that no aiding has the most consistent egomotion error distance (consistently poor). When the robot moves straight (i.e. no aiding), it identifies mostly the same features for each run and hence, egomotion variation is small. Non evaluative action aiding has the largest variation for egomotion error distance. The robot turns at positions that are not fixed in each test run. At times, the robot heads in directions with favorable landmark positions while at other instances, landmark positions may be poor. This variability causes the large variation in egomotion error distance for non evaluative action aiding technique. In the simulated evaluative action aiding scheme, the algorithm evaluates the best orientation for each robot step. As the robot travels down the same corridor, the algorithm identifies similar landmarks. Given inaccuracies in robot motions (especially since robot movements were manually controlled in the simulated test), evaluative action aiding has larger egomotion error distance variations compared to no aiding, but smaller egomotion error distance variation compared to non evaluative action aiding. Table 9 summarizes the average egomotion error distance variation for the various action aiding techniques.

Table 9: Table summarizing the average (for tests from location 1 to 2 and from location 2 to 1) egomotion error distance standard deviation for different action aiding types.

	No aiding (long route)	Non evaluative action aiding (long route)	Simulated evaluative action aiding
Average Std Dev (m)	1.6m	2.6m	2.3m

4.5.3 Successful egomotion operation in area with few features

In the limited feature area test, non evaluative action aiding could not be completed for every single test as egomotion fails expectedly when the robot faces the featureless wall. However, since evaluative action aiding has the capability to determine the location of available landmarks and positions the robot to use these landmarks for egomotion calculations, egomotion did not fail in any tests, thus successfully operating in an area with few features. With evaluative action aiding, egomotion error distance was reduced by 76% compared to no aiding (Table 10). This test results demonstrates the successful accomplishment of this research's second goal: to develop an action aiding technique that reduce egomotion error distance and increases the usability of egomotion in areas with few features.

In large areas with a lot of well distributed landmarks, sufficient landmarks would have been identified even when the robot is travelling straight. Hence, evaluative action aiding is not expected to give the large egomotion accuracy improvements seen in this research. This is not validated in this research and could be assessed in future work.

Table 10: Average egomotion distance errors and improvement percentage (compared to the egomotion error distance from no aiding) for evaluative action aiding in limited features areas.

	No aiding average error (baseline)	Simulated evaluative action aiding	
Average egomotion error	9.2m	2.2m	76% improvement

4.5.4 Egomotion errors distances and standard deviation increase with increased landmark distances

When the robot moves from location 1 to 2 and from location 2 to 1, different egomotion error distance results are produced for the same test (i.e. no aiding, non evaluative aiding and simulated evaluative aiding) even though the robot moves through the same distances (Table 11). Moving towards location 2 gives a higher average egomotion error distance (for all movement profiles) compared to moving towards location 1. Likewise, it is also seen that moving towards location 2 gives a larger egomotion error distance standard deviation. When the robot moves towards location 2, it "sees" a longer corridor (Section 4.4.2) even though the physical distance moved is the same. With a longer corridor, landmarks are farther giving more measurement uncertainties resulting in greater egomotion error distances and standard deviation.

Table 11: Summary of the egomotion error distances and standard deviation as the robot moves from location 1 to 2 and from location 2 to 1.

	Location 1 to 2		Location 2 to 1	
	Average error	**Std Dev**	**Average error**	**Std Dev**
No aiding (baseline)	14.0m	1.8m	8.6m	1.4m
Non evaluative action aiding	10.0m	3.4m	5.7m	1.9m
Simulated evaluative action aiding	3.9m	3.1m	2.2m	1.6m

V. Conclusions and Recommendations

This chapter concludes the research and highlights its significance. Thereafter, recommendations are made for future work and potential applications.

5.1 Research conclusion

This research successfully developed action aiding algorithms that reduce egomotion distance errors by an average of 31.5% (non evaluative action aiding) and 72.5% (evaluative action aiding with landmark tracking), therefore meeting the first goal in this research. Notably, evaluative action aiding enables reliable use of egomotion in an area with few features (achieving the second goal of this research) with none of the egomotion tests in the limited feature area failing. Evaluative action aiding achieved a 76% reduction of egomotion distance errors in the limited feature area test. Since action aiding is based on the concept of external repositioning of the robot and not extensive concept change to image navigation / egomotion, the action aiding techniques can be easily applied to all image navigation solutions with minimal modifications to existing applications (meeting the third goal of this research). Action aiding enhances egomotion reliability and accuracy, potentially allowing standalone image navigation operations with improved precision compared to existing image navigation performances. Action aided image navigation could be suitable for small, less complex, low powered robots with space and power limitations.

5.2 Significance of Research

Much effort has gone into improving image navigation accuracy. It is also known that areas with few features cause significant egomotion errors. However, this research has demonstrated that significant improvements on image navigation and egomotion accuracy are possible with action aiding. The performance improvement means that image navigation can potentially be accurate enough for standalone use without

augmentation from other navigation systems. If used with other navigation system augmentations, the increased egomotion accuracy will improve the overall navigation package. It is also demonstrated that action aiding techniques can be implemented with existing image navigation system with minimal effort using a simple, light weight action aiding engine that needs no modification to existing hardware, and very little integration efforts to all the existing image navigation software.

5.3 Recommendations for future research

The following recommendations on possible future research areas are proposed. This is based on the experiences and ideas gained in this project.

5.3.1 Use of landmark height information in evaluation

For simplification, the present action aiding algorithm does not include the landmark height effects on egomotion errors in its evaluation of the robot orientations for the next step. It is based on the 2D location (horizontal distance and depth) information. However, height also plays a similar effect, with landmarks near the camera level giving the largest egomotion errors. This concept can be extended to 3D space where the landmark location in the whole image space is considered in the evaluation. It is expected that a more accurate evaluation of the robot orientation is possible giving farther reduction of egomotion errors if all aspects of the landmark (distance, horizontal displacement, height) are considered.

5.3.2 Implement full egomotion algorithm

The full egomotion algorithm should be incorporated into the present post processing algorithm. Thereafter, robot movements do not need to be decoupled and can be tested with more realistic operational movements. The actual potential of the action aiding profiles on egomotion errors can then be determined.

5.3.3 Real-time egomotion to sense robot movements

Currently, the robot odometry is used to measure robot movements. However, odometry present many inaccuracies. For example, when the robot starts to move forward, one wheel may start off slightly faster than the other, causing a slight slew. Calibration inaccuracies can also cause one wheel to turn slightly faster. Unbalanced left and right wheel tire pressures can farther cause the robot to veer to a side. All these while, the onboard odometry interprets the robot as moving straight. Inaccurate odometry readings cause the robot to move to angles / positions that are not commanded by the action aiding algorithm, reducing the effectiveness of action aiding in reducing egomotion errors. During the physical test of the various action aiding techniques using the robot, odometry errors impacted the successful conduct of autonomous navigation with evaluative action aiding (landmark tracking) test.

If egomotion can be incorporated in the robot and measures robot movements in real time, it could be possible not to use odometry to measure robot movements. Inaccuracies associated with odometry measurements can be avoided. However, real-time image processing is computing resources intensive and hence, dedicated GPUs for image processing are required if real time egomotion is desired. The robot would also have to travel slower to allow computation to complete.

5.3.4 Steerable cameras

When landmarks are located at the desired regions, egomotion accuracy is improved. One key reason the robot turns for action aiding is because the camera system used is a pair of fixed (i.e. unmovable) camera. Hence the robot needs to physically re-orientate itself to effectively re-orientate the landmark image locations. If the robot is equipped with steerable cameras, the cameras can be slaved instead of physically orientating the robot. This re-orientation of the cameras can be done continuously while the robot moves, which reduces the journey time. Farthermore, accuracy is expected to be

farther increased as optimization can be done continuously, and also the cameras can be slaved with more precision compared to the physical robot.

5.3.5 Landmark tracking using 2 pairs of steerable cameras

Another approach is to slave one pair of steerable camera to track a single cluster of landmarks (located at favorable positions) as the robot moves. The resulting camera movement can be measured and converted to robot movements. As the camera movements can be measured more precisely, the resulting egomotion errors should be small. The other pair of camera will search for the next set of landmarks to use.

5.3.6 Use of side images

The landmark position egomotion error model implies that if a landmark is located perpendicular to the robot's direction of travel (i.e. the image is located to the side of the moving robot), use of this landmark to calculate egomotion gives the least egomotion error. Intuitively, side-located landmark image displacement is the greatest (projected in the side cameras) compared to all other positions and hence for the same sensor noise, the egomotion calculation signal strength (P_S) is greatest, hence the largest SNR, giving the least egomotion errors. The robot will need to be equipped with side facing cameras [30].

5.3.7 360° view

When a robot navigates in a small enclosed environment, the robot has a high probability of being physically close to a featureless wall, resulting in the failure of egomotion when no landmarks are identified. If 360° view cameras are used, the chance of finding landmarks in the environment is greatly increased. Farthermore with increased views, more landmarks can be tracked simultaneously and combined using weighted averaging for increased egomotion accuracy.

5.4 Summary

In standalone mode, image navigation often has poor accuracies especially if it operates in areas with few landmarks such as along long narrow corridors. Studies exist to reduce egomotion errors through approaches such as removal of feature outliers, improving feature matching etc. There are also studies to integrate image navigation with other navigation systems to produce an integrated navigation package. Existing research efforts show that landmark locations affect the accuracy of landmark distance measurements. Therefore this research proposes that there is a direct relation between landmark locations and egomotion accuracy. There are desired regions (relative to the robot) for landmarks to be located at that gives the least egomotion errors if used for egomotion calculation. This research hypothesized that if the robot orientates (i.e. action aiding) so that landmarks are positioned at the desired image regions, egomotion errors are reduced (compared to no aiding). It was also predicted that action aiding will improve image navigation in areas with few landmarks, by actively seeking available landmarks and placing these landmarks in favorable image positions.

Firstly, this research proved that landmark measurement variations exist and systematically showed the relation between landmark locations and their effects on egomotion errors. A landmark location egomotion error model is then formed. Based on the error model CV distribution for different landmark positions, three action aiding techniques are proposed: (1) non evaluative action aiding, (2) evaluative action aiding (physical scan), and (3) evaluative action aiding (landmark tracking). These proposed techniques are first evaluated in a simulation environment to understand the different techniques' behavior and to compare their relative performances against no aiding. Simulation results show the reduction of the journey egomotion CV when the robot journey are action aided; no aiding has the largest journey egomotion CV, followed by non evaluative action aiding, with evaluative action aiding having the least journey egomotion CV.

The action aiding algorithm is next developed for implementation in the robot for physical testing. Results show what simulation had predicted; when the robot is tested without any form of action aiding, it has the largest egomotion error distance. Non evaluative action aiding reduces egomotion error distances while evaluative action aiding (landmark tracking) provides the largest egomotion error distance reduction. The limited feature area test also shows that action aiding enables egomotion to operate reliably and with reduced egomotion errors compared to no action aiding. In summary, research goals are accomplished.

Appendix A – Detailed block diagram of action aiding processes

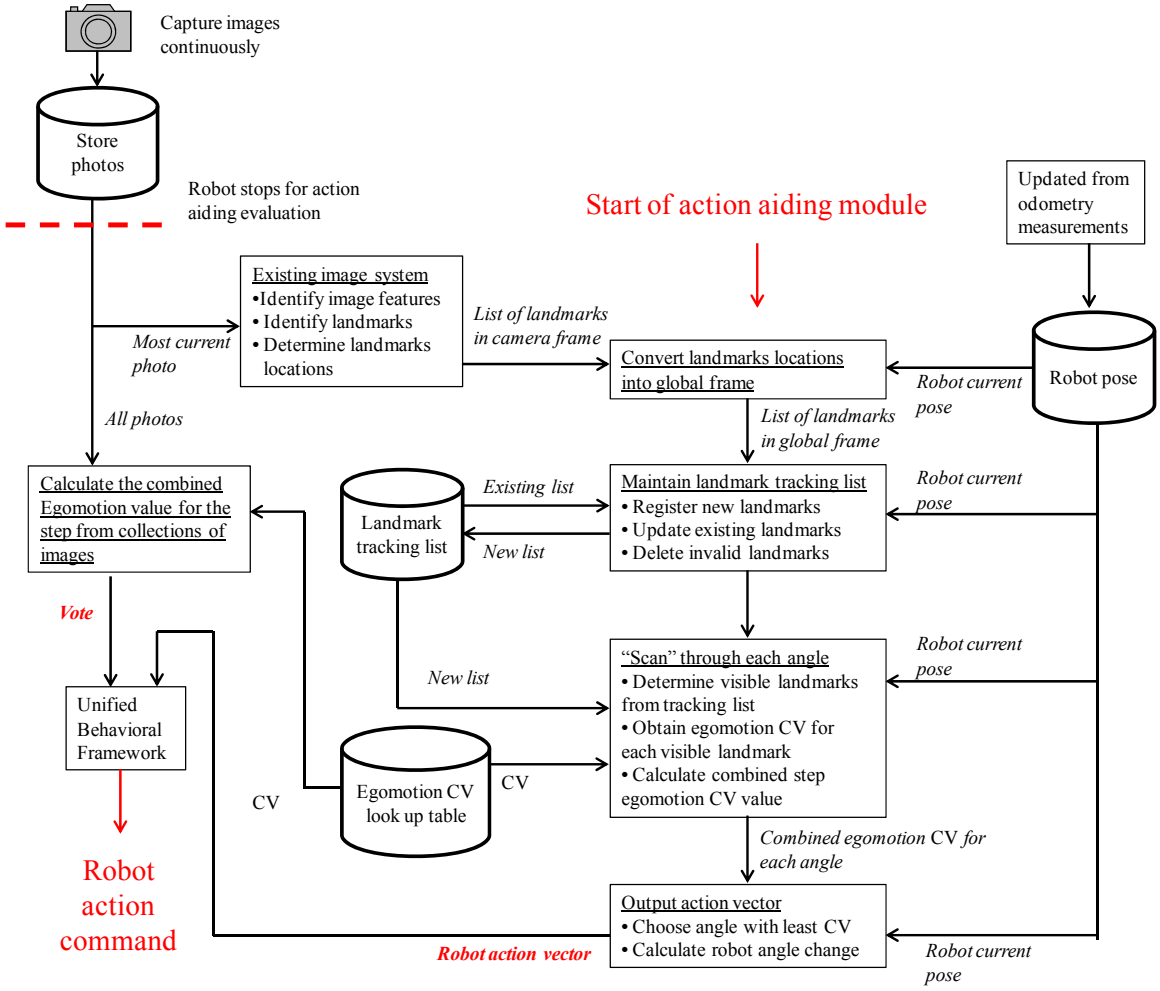

Figure 42: Detailed block diagram of action aiding processes.

Bibliography

[1] David G Hoag, "Apollo Guidance, Navigation and Control," MIT Instrumentation laboratory, Progress report presented at the National Space Meeting of the Institute of Navigation April 22-24, 1969, Houston Texas 1969.

[2] Matthias O Franz and Hanspeter A Mallot, "Biomimetic robot navigation," *Robotics and Autonomous Systems*, vol. 30, pp. 133-153, 2000.

[3] Sara Susca. (2010, July) GPS world. [Online]. http://www.gpsworld.com/tech-talk-blog/gnss-independent-navigation-solution-using-integrated-lidar-data-11378

[4] Kurt Konolige, Motilal Agrawal, and Joan Sola, "Large Scale Visual Odometry for Rough Terrain," Stanford Research Institute (SRI) International, CA, The report documents the work performed as part of DARPA's Learning Applied to Ground Robots.

[5] Eagle Sunrise Jones, "Large Scale Visual Navigation and Community Map Building," University of California, Los Angeles, Dissertation 2009.

[6] Enric Celaya, José Luis Albarral, and Tom Creemers, "A flexible platform for vision based robot navigation," in *Proceedings of the first International Workshop on Telerobotic and Augmented Reality for Teleoperation*, Universidad Politécnica de Madrid, Madrid, Espanya, 2005, pp. 1-22.

[7] Clark F Olson, Larry H Matthies, Marcel Schoppers, and Mark W Maimone, "Rover navigation using stereo ego-motion," *Robotics and autonomous Systems*, vol. 43, no. 4, pp. 215-229 (15), June 2003.

[8] Michael Milford, David McKinnon, Michael Warren, Gordon Wyeth, and Ben Upcroft, "Feature-based Visual Odometry and Featureless Place Recognition for SLAM in 2.5D environments," in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA)*, Australian Robotics & Automation Association , Monash University, Melbourne, 2011, pp. 1-8.

[9] Christopher P. Weyers, "Multiple Integrated Navigation Sensors For Improved Occupancy Grid FastSLAM," Air Force Institute of technology (AFIT), Dayton, MSc Thesis 2011.

[10] Clark F Olson, Larry H Matthies, Marcel schoppers, and Mark W Maimone, "Robust Stereo Ego-motion for Long Distance Navigation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, Hilton Head Island, SC, 2000, pp. 453 - 458.

[11] Anderson S Claus and Madsen B Claus, "Optimal Landmark Selection For Triangulation of Robot Position," *Robotics and Autonomous Systems*, vol. 23, no. 4, pp. 277–292, June 1998.

[12] Mitch Bryson and Salah Sukkarieh, "Observability Analysis and Active Control for Airborne SLAM," *IEEE Transactions on Aerospace and electronics Systems*, vol. 44, no. 1, pp. 261-280, January 2008.

[13] Cyrill Stachniss, Dirk Hahnel, and Wolfram Burgard, "Exploration with Active Loop-Closing for FastSLAM," in *Proceedings of the Intelligent Robots and Systems (IROS)*, 2004, pp. 1505 - 1510 vol.2.

[14] Michael J. Veth, "Fusion of Imaging and Inertial Sensors for Navigation," Air Force Institute of Technology (AFIT), Dayton, PhD Dissertation 2006.

[15] Peter S Maybeck, *Stochastic Models, Estimation, and Control*.: Academic Press, Inc (copyright now owned by Navtech Seminars & GPS Supply), 1979, vol. 1.

[16] George Casella and Roger L Berger, *Statistics Inference*, 2nd ed.: Duxbury Advanced Series, 2002.

[17] Javed H Niazi. (2009, Oct) Error analysis in biosensing. Sabanci University Faculty of Engineering & Natural Sciences BIO 580 lecture slides. [Online]. http://myweb.sabanciuniv.edu/javed/files/2009/10/Week-14.pdf

[18] Jia Li. (2009, Feb) The Pennsylvania State University. [Online]. http://sites.stat.psu.edu/~jiali/course/stat416/notes/chap2.2.pdf

[19] Annalisa Milella and Roland Siegwart, "Stereo-Based Ego-Motion Estimation Using Pixel Tracking and Iterative Closest Point," in *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006)*, 2006, p. 21.

[20] David G Lowe, "Object Recognition from Local Scale-Invariant Features," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, vol. 2, Kerkyra, 1999, pp. 1150 - 1157.

[21] George Bebis. Class Presentation, CS485/685 Computer Vision, Interest Point Descriptors and Matching. Powert Point Presentation. [Online]. http://www.cse.unr.edu/~bebis/CS485/Lectures/InterestPointDescriptorsMatching.ppt

[22] Stephen Se, David Lowe, and Jim Little, "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *The International Journal of Robotics Research*, vol. 21, pp. 735-758, 2002.

[23] Ashutosh Saxena, Min Sun, and Andrew Y Ng, "Make3D: Depth Perception from a Single Still Image," in *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, Chicago, Aug. 2008, pp. 1571-1576.

[24] Armangue Xavier, Araujo Helder, and Salvi Joaquim, "A review on egomotion by means of differential epipolar geometry applied to the movement of a mobile robot," *Pattern Recognition*, vol. 36, no. 12, pp. 2927-2944, Dec. 2003.

[25] Takashi Izumi, Yoshiharu Ueki, and Yutaka Takahashi, "Height Measurement System using Stereo Vision," in *Proceedings of the SICE Annual Conference (Instrumentation, Control, Information Technology and System Integration)*, Fukui University, Japan, 2003, pp. 719 - 724.

[26] Tina Y Tian, Carlo Tomasi, and David J Heeger, "Comparison Of Approaches To Egomotion Computation," in *Proceedings of the Computer Vision and Pattern Recognition Conference (1996)*, San Francisco, CA, 1996, pp. 315 - 320.

[27] Dani¨el Fontijne, Leo Dorst, Frans C.A. Groen, and Wannes Vandermark, "Vehicle Ego-Motion Estimation with Geometric Algebra," in *Proceedings of the Intelligent Vehicle Symposium, IEEE*, vol. 1, 2002, pp. 58 - 63.

[28] Brian G Woolley, "Unified Behavorial Framework for Reactive Robot Control in Real-Time Systems," Air Force Institute of Technology (AFIT), Dayton, MSc Thesis 2007.

[29] Feng Lu and Evangelos Milios, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans," in *Proceedings of the Computer Vision and Pattern Recognition*, 1994, pp. 935 - 938.

[30] Juan D Jurado, "Enhanced Image-aided Navigation Algorithm with Automatic Calibration and Affine Distortion Prediction," Air Force Institute of Technology (AFIT), Dayton, Thesis 2012.

Vita

ME4 (Captain) Eng Kwee Guan is a Republic of Singapore Air Force (RSAF) Officer. He graduated from the Singapore Polytechnic in 1999 with a Diploma with Merit in Electrical, Computer and Communications Engineering and was also awarded the Ilog book prize. In 2002, he was commissioned as an RSAF officer and held appointments in the RSAF operational unit and in HQ as a staff officer before being awarded the Singapore Armed Forces (SAF) Academic Training Award (ATA) for undergraduate studies at Imperial College London. He graduated with a BEng in Electrical Engineering and finished in the top 3 in his graduating cohort. He was also awarded Usmani prize in micro-electronics and was the top 3 finalist at U.K's "Best Electronic Engineering Student" 2008 Science, Engineering & Technology (SET) student of the year competition. He held another appointment as an avionics staff officer at the RSAF HQ before being awarded a SAF postgraduate award for postgraduate studies at Air Force Institute of Technology (AFIT). During this time, he was invited to become a member at both Tau Beta Pi and Eta Kappa Nu engineering society. When he returns back to the RSAF, he will assume appointment as an avionics branch section head.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
13-09-2012	Master's Thesis	3 Oct 11 – 13 Sep 12

4. TITLE AND SUBTITLE

Intelligent Behavioral Action Aiding for Improved Autonomous Image Navigation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Eng, Kwee G., ME4 (Captain), Republic of Sigapore Air Force

5d. PROJECT NUMBER
12G194

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/12-46

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory, Sensors Directorate, Reference Branch
Attn: Dr. Jacob Campbell
2241 Avionics Circle
WPAFB OH 45433-7765
(937) 255-6127, x4154 (DSN: 785-6127, x4154) jacob.campbell@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RYRN

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and Singapore Government and is not subject to copyright protection in the United States and Singapore.

14. ABSTRACT

In egomotion image navigation, errors are common especially when traversing areas with few landmarks. Since image navigation is often used as a passive navigation technique in Global Positioning System (GPS) denied environments; egomotion accuracy is important for precise navigation in these challenging environments. One of the causes of egomotion errors is inaccurate landmark distance measurements, e.g., sensor noise. This research determines a landmark location egomotion error model that quantifies the effects of landmark locations on egomotion value uncertainty and errors. The error model accounts for increases in landmark uncertainty due to landmark distance and image centrality. A robot then uses the error model to actively orient to position landmarks in image positions that give the least egomotion calculation uncertainty. Two actions aiding solutions are proposed: (1) qualitative non-evaluative aiding action, and (2) quantitative evaluative aiding action with landmark tracking. Simulation results show that both action aiding techniques reduce the position uncertainty compared to no action aiding. Physical testing results substantiate simulation results. Compared to no action aiding, non-evaluative action aiding reduced egomotion position errors by an average 31.5%, while evaluative action aiding reduced egomotion position errors by an average 72.5%. Physical testing also showed that evaluative action aiding enables egomotion to work reliably in areas with few features, achieving 76% egomotion position error reduction compared to no aiding.

15. SUBJECT TERMS
Action aiding, egomotion, landmark position effects on egomotion error, egomotion in feature scarce areas

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			Dr. Gilbert L. Peterson (ENG)
U	**U**	**U**	**UU**	140	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, x4281 (gilbert.peterson@afit.edu)